



**Centro Universitário de Brasília – UniCEUB**  
**Faculdade de Exatas e Tecnologia - FAET**  
**Curso de Engenharia da Computação**

**Mannuela Cruz Vidigal de Oliveira**

# **Sistema Automatizado de Pedidos e Serviços**

**Brasília**  
**2007**

**Mannuela Cruz Vidigal de Oliveira**

# **Sistema Inteligente para Automação de Pedidos e Serviços**

Trabalho apresentado ao Centro Universitário de Brasília (UNICEUB) Como pré-requisito para a obtenção de Certificado de Conclusão do Curso de Engenharia da Computação.

Orientador: Prof. José Julimá Bezerra Junior

**Brasília**

**2007**

# **AGRADECIMENTOS**

Dedico este projeto primeiramente à Deus por ter me concedido a graça de poder estudar e ter a capacidade de aprender, ler, pensar e agir. Aos meus pais que nunca duvidaram de mim e sempre me apoiaram nas minhas decisões, em especial ao meu pai que fez engenharia e me incentivou na vontade de segui-lo. Aos meus amigos da Engenharia que estiveram presentes nos grupos de estudos promovidos e nas demais situações diárias de minha vida. Ao meu noivo que esteve ao meu lado no término do curso me dando total apoio para a conclusão do mesmo. Aos professores que me incentivaram e estavam sempre dispostos a tirar dúvidas, auxiliar na resolução de problemas e mostrar o caminho de busca para ter um melhor conhecimento.

# SUMÁRIO

<b>AGRADECIMENTOS.....</b>	<b>I</b>
<b>SUMÁRIO.....</b>	<b>II</b>
<b>LISTA DE FIGURAS.....</b>	<b>IV</b>
<b>LISTA DE TABELAS.....</b>	<b>VI</b>
<b>LISTA DE TABELAS.....</b>	<b>VI</b>
<b>LISTA DE ABREVIATURAS.....</b>	<b>VII</b>
<b>RESUMO.....</b>	<b>VIII</b>
<b>ABSTRACT.....</b>	<b>IX</b>
<b>CAPÍTULO 1 - INTRODUÇÃO.....</b>	<b>1</b>
1.1 MOTIVAÇÃO.....	1
1.2 OBJETIVOS.....	1
1.3 METODOLOGIA DE PESQUISA.....	2
1.4 ESTRUTURA DA MONOGRAFIA.....	2
<b>CAPÍTULO 2 – REFERENCIAL TEÓRICO.....</b>	<b>3</b>
2.1 AUTOMAÇÃO.....	3
2.2 SERVIDOR WEB.....	3
2.2.1 APACHE.....	4
2.2.2 NAVEGADOR.....	5
2.3 BANCO DE DADOS.....	6
2.3.1 MYSQL.....	6
2.4 LINGUAGENS DE PROGRAMAÇÃO PARA WEB.....	7
2.4.1 HTML.....	7
2.4.2 XHTML.....	7
2.4.3 CSS.....	7
2.4.4 AJAX.....	8
2.4.5 PHP.....	8
2.5 COMPONENTES FÍSICOS.....	9
2.5.1 COMPUTADOR PESSOAL.....	9
2.5.2 PORTA PARALELA.....	10
2.5.3 PONTO-DE-ACESSO.....	10
2.5.4 DISPLAY.....	11
2.5.5 PDA.....	11
<b>CAPÍTULO 3 – DESENVOLVIMENTO.....</b>	<b>13</b>
3.1 CONCEITOS INICIAIS.....	13
3.1.1 INTERAÇÕES COM OS USUÁRIOS.....	13
3.1.2 MODELAGEM DO SISTEMA.....	17
3.1.3 MODELAGEM DO BANCO DE DADOS.....	19

3.1.4 IMPLEMENTAÇÃO DO SISTEMA .....	20
<b>3.2 CUSTO PARA IMPLEMENTAÇÃO.....</b>	<b>24</b>
<b>3.3 MÓDULOS .....</b>	<b>25</b>
3.3.1 CAIXA.....	25
3.3.2 BALCÃO .....	30
3.3.2.1 Cozinha .....	33
3.3.3 CLIENTE .....	34
<b>3.4 INTEGRAÇÃO .....</b>	<b>39</b>
3.4.1 CAIXA / CLIENTE .....	39
3.4.2 CAIXA / BALCÃO.....	40
3.4.3 BALCÃO / COZINHA .....	41
<b>CAPÍTULO 4 – CONCLUSÃO.....</b>	<b>46</b>
<b>4.1 FUTUROS PROJETOS .....</b>	<b>46</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>48</b>
<b>SITES .....</b>	<b>48</b>
<b>APÊNDICE I .....</b>	<b>52</b>
<b>APÊNDICE II.....</b>	<b>208</b>
<b>APÊNDICE III .....</b>	<b>211</b>

# LISTA DE FIGURAS

FIGURA 1 - DIAGRAMA DE SERVIDOR WEB .....	4
FIGURA 2 - DIAGRAMA DE CASO DE USO .....	14
FIGURA 3 - CASO DE USO: REALIZAR LOGIN .....	14
FIGURA 4 - CASO DE USO: REALIZAR PEDIDO .....	15
FIGURA 5 - CASO DE USO: SOLICITAR CONTA.....	15
FIGURA 6 - CASO DE USO: REALIZAR LOGIN (PERFIL BALCÃO) .....	15
FIGURA 7 - CASO DE USO: VISUALIZAR PEDIDO .....	16
FIGURA 8 - CASO DE USO: LIBERAR PEDIDO .....	16
FIGURA 9 - CASO DE USO: REALIZAR LOGIN (PERFIL CAIXA).....	16
FIGURA 10 - CASO DE USO: GERENCIAR GARÇOM.....	17
FIGURA 11 - CASO DE USO: GERENCIAR CONTA .....	17
FIGURA 12 - CASO DE USO: ENCERRAR MESA .....	17
FIGURA 13 - DIAGRAMA DE ARQUITETURA .....	18
FIGURA 14 - MODELO DE ENTIDADE-RELACIONAMENTO .....	20
FIGURA 15 – DIRETÓRIO PRINCIPAL .....	20
FIGURA 16 – ESTRUTURA DA PASTA APACHE GROUP .....	21
FIGURA 17 – ESTRUTURA DA PASTA WWW.....	22
FIGURA 18 - DIAGRAMA DE CLASSES .....	23
FIGURA 19 – TELA DE LOGIN.....	25
FIGURA 20 – TELA PRINCIPAL - CAIXA .....	26
FIGURA 21 – MESA OCUPADA .....	27
FIGURA 22 – COORDENA GARÇOM .....	28
FIGURA 23 – ENCERRAR CONTA .....	29
FIGURA 24 – TELA DE CONTROLE .....	29
FIGURA 25 – FLUXOGRAMA DO CAIXA .....	30
FIGURA 26 – NOVO PEDIDO .....	31
FIGURA 27 - TELA DE CONTROLE DE PEDIDO.....	32
FIGURA 28 – FLUXOGRAMA DO BALCÃO.....	33
FIGURA 29 – TELA DE LOGIN DO CLIENTE.....	35
FIGURA 30 – TELA DE CADASTRO.....	35
FIGURA 31 – TELA DE CARDÁPIO.....	36
FIGURA 32 – TELA DE VALORES.....	36
FIGURA 33 – CONTA PARCIAL.....	37
FIGURA 34 – CONTA FECHADA .....	38
FIGURA 35 – FLUXOGRAMA DO CLIENTE.....	39
FIGURA 36 – ARQUIVO TEXTO.....	42
FIGURA 37 – TELA DE CONFIGURAÇÃO DO SMARTIE .....	43

<b>FIGURA 38 – SIMULADOR DO SMARTIE .....</b>	<b>43</b>
<b>FIGURA 39 – FLUXOGRAMA DE INTERAÇÃO .....</b>	<b>44</b>
<b>FIGURA 40 – IMPLEMENTAÇÃO FINAL .....</b>	<b>45</b>

# LISTA DE TABELAS

TABELA 1 - COLABORADORES ATIVOS .....	5
TABELA 2 - TOTAL PARA USUÁRIOS ATIVOS ATRAVÉS DE TODOS OS DOMÍNIOS.....	5



# LISTA DE ABREVIATURAS

- AJAX** – *Asynchronous Javascript and XML* – JavaScript Assíncrono e XML
- CSS** – *Cascading Style Sheets* – Folha de Estilos em Cascata
- DNS** – *Domain Name System* – Sistema de Nomes de Domínio
- HTML** – *Hypertext Markup Language* – Linguagem de Marcação de Hipertexto
- HTTP** – *HyperText Transfer Protocol* – Protocolo de Transferência de Hipertexto
- IP** – *Internet Protocol* – Protocolo de Internet
- LCD** – *Liquid Cristal Display* – Display de Cristal Líquido
- PC** – *Personal Computer* – Computador Pessoal
- PDA** – *Personal Digital Assistant* – Assistente Pessoal Digital
- PHP** – *PHP Hypertext Preprocessor* – Preprocessador de Hipertexto
- SGBD** – Sistema Gerenciador de Banco de Dados
- SisMannu** – Sistema da Mannu
- SQL** – *Structure Query Language* – Linguagem Estrutura de Consulta
- TI** – Tecnologia da Informação
- USB** – *Universal Serial Bus* – Barramento Serial Universal
- W3C** – *World Wide Web Consortium* – Consórcio da Rede de Alcance Mundial
- WWW** – *World Wide Web* – Rede de Alcance Mundial
- XHTML** – *eXtensible Hypertext Markup Language* – Linguagem Extensível para Marcação de Hipertexto
- XML** – *eXtensible Markup Language* – Linguagem Extensível de Marcação

# RESUMO

Neste projeto é implementado um sistema inteligente para automatização de pedidos e serviços em estabelecimentos, tais como restaurantes e lanchonetes. A automação comercial foi implementada com soluções de custo acessível e com interfaces de fácil manuseio por qualquer usuário. A estrutura do projeto é formada por um servidor que controlará o envio de informações entre o cliente, a cozinha e o caixa do estabelecimento.

Palavras-Chave: automação comercial; PHP; custo acessível; PDA; *Open Source*; *Browser*; LCD

# ABSTRACT

The objective of this project is to implement a intelligent system to automate the services and orders of a place like restaurants and snack bars. The commercial automation was implemented with low cost solutions and easy interface for all user. The structure of the project consists of a server that controls the sending of data between the costumer, the kitchen and the checkout of the place.

Keywords: commercial automation; PHP; low cost; PDA; *Open Source*; *Browser*; LCD

# **CAPÍTULO 1 - INTRODUÇÃO**

## **1.1 MOTIVAÇÃO**

Cada vez mais procura-se “facilitar” as atividades no dia-a-dia das pessoas e encontrar soluções para a obtenção de um melhor resultado. O avanço tecnológico tem gerado um grande empenho e interesse no desenvolvimento de sistemas para automatizar serviços, ou seja, há uma busca enorme por parte de empresas e fornecedores de serviços, por soluções “práticas”, automatizadas e com um custo acessível para um melhor desempenho de suas tarefas.

A interoperabilidade dos serviços fez surgir a busca pela melhoria da logística funcional em diversas áreas de atuação, como por exemplo, a alimentícia. Desta forma, o cliente, pode transportar o ambiente agradável de sua casa para ter a mesma liberdade no local físico determinado.

A falta de tempo com os compromissos diários faz com que as pessoas tenham como opção a alimentação em um desses estabelecimentos. A exatidão do pedido e a garantia da entrega no tempo correto tornam-se então inerentes no desenrolar das atividades desenvolvidas no dia.

Além dos motivos citados, faz-se necessário no mundo atual uma integração das pessoas com soluções tecnológicas. Com esta solução, diversas pessoas poderão sentir-se motivadas a buscar o mundo digital e acabar com o paradigma de que a “máquina é um bicho de sete cabeças” com o simples ato de se alimentar.

## **1.2 OBJETIVOS**

O objetivo do projeto é implementar um sistema de automação comercial alimentícia pela sinergia de soluções tecnológicas. O sistema controlará as funcionalidades do estabelecimento, tais como: os pedidos feitos pelos clientes, o recebimento e execução pela cozinha e o valor a ser pago no caixa; garantindo assim uma otimização e eficiência na execução das tarefas envolvidas.

Paralelamente ao objetivo acima citado, busca-se alcançar um melhor índice na inclusão digital em nossa sociedade, através da capacitação e interação do usuário final, cliente e funcionário.

### 1.3 METODOLOGIA DE PESQUISA

Este projeto segue uma linha de pesquisa bibliográfica baseada em consultas a materiais publicados em livros, periódicos e sites da Internet. Com base nestas pesquisas o projeto tem finalidade aplicada, tendo com isso aplicação prática e experimental de seu objeto de estudo. [SANTOS, 2000]

### 1.4 ESTRUTURA DA MONOGRAFIA

Esta monografia foi dividida em quatro capítulos:

No Capítulo 1 é apresentada a proposta do projeto com os objetivos e motivação para o desenvolvimento.

No Capítulo 2 é apresentado o referencial teórico com as definições de cada tecnologia utilizada no trabalho e suas respectivas aplicações ao longo do projeto.

No Capítulo 3 é apresentado o desenvolvimento do projeto com um todo, descrevendo a implementação do software em seus módulos e do hardware integrado. É o capítulo que mostra o funcionamento do sistema na logística do estabelecimento.

No Capítulo 4 é apresentada a conclusão do projeto e sugestões para futuras evoluções do mesmo.

## CAPÍTULO 2 – REFERENCIAL TEÓRICO

### 2.1 AUTOMAÇÃO

Automação (do inglês *Automation*) é um sistema automático de controle pelos quais os mecanismos verificam seu próprio funcionamento, efetuando medições e introduzindo correções, sem a necessidade da interferência do homem. [HOLANDA, Aurélio Buarque de]

Hoje em dia está presente em diferentes níveis de atividades do homem, desde a medicina até a astronomia, ampliando a capacidade de interação com a natureza e os processos.

A automação está presente no projeto permitindo que as principais atividades como Fazer Pedido e Encerrar Conta são feitos pelo computador em um sistema que integra diversas tecnologias.

### 2.2 SERVIDOR WEB

O servidor web é um programa que recebe um pedido HTTP feito por um *browser* e esse pedido é composto por um domínio no qual é traduzido pelo DNS para seu respectivo IP. Ao receber o pedido, o servidor procura os documentos relativos ao mesmo e retorna o HTML via protocolo HTTP ao *browser* (**Figura 1**).

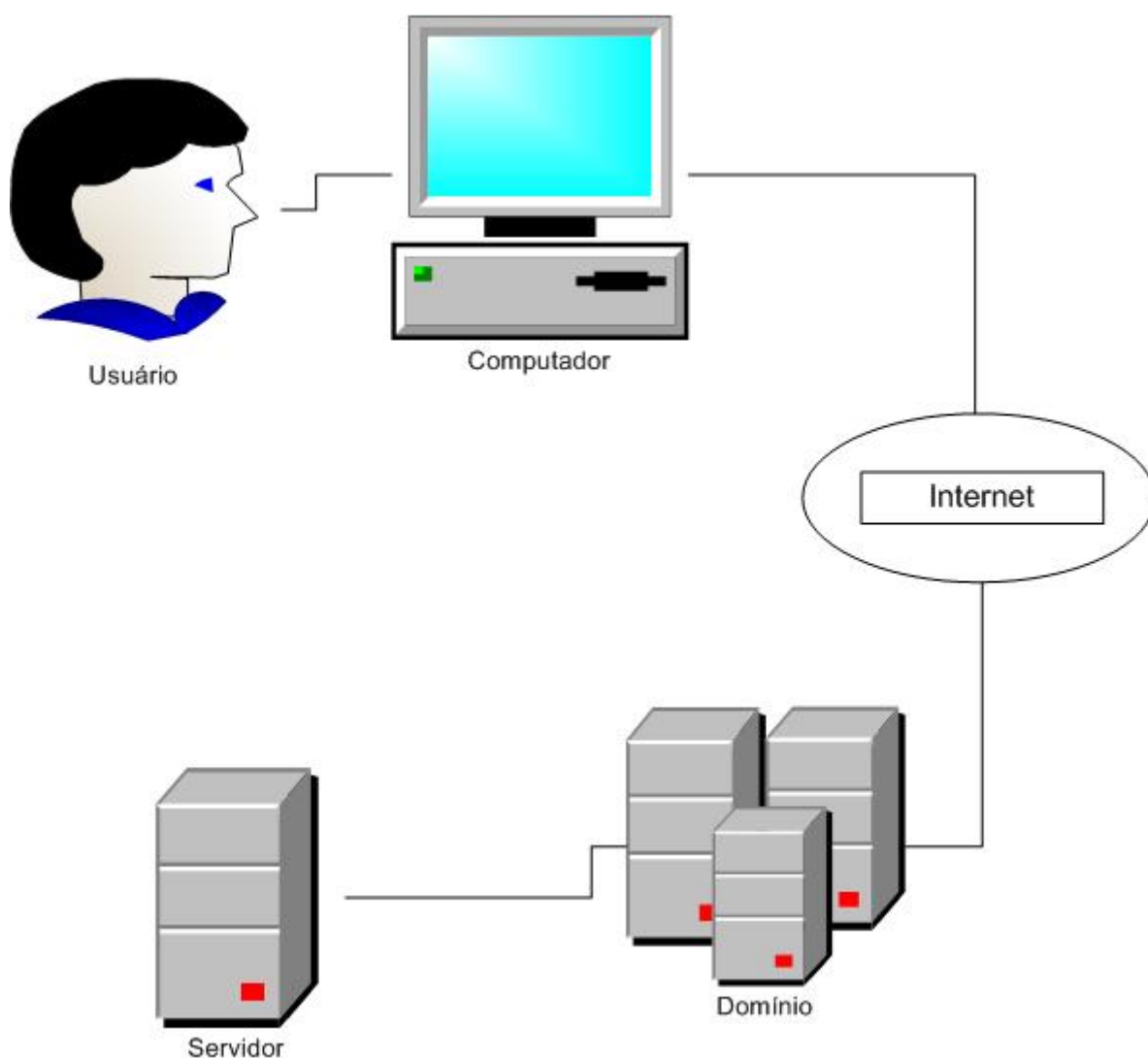


Figura 1 - Diagrama de Servidor Web

### 2.2.1 Apache

O Projeto Apache é um esforço coletivo de vários colaboradores, para um desenvolvimento de um software robusto, gratuito, e com qualidade, para a implementação de um servidor HTTP (*HyperText Transfer Protocol*) – altamente usado na internet. O projeto é administrado por um grupo de voluntários localizados no mundo todo, que se comunicam através da internet, para planejar, desenvolver o Apache e sua documentação. [INFOWESTER, 2006]

Apache tem sido o servidor web mais popular na Internet desde 1996. As pesquisas feitas pelo *Netcraft* mostram que 70% dos sites disponíveis na internet utilizam o Apache. [NETCRAFT, 2006]

**Tabela 1 - Colaboradores Ativos**

Colaborador	Março 2006	Por cento	Abril 2006	Por cento	Mudança
Apache	53287298	68,70	50588433	62,72	- 5,98
Microsoft	15912427	20,51	20343656	25,22	4,71
Sol	1881587	2,43	1907503	2,36	- 0,07
Zeus	574607	0,74	563381	0,70	- 0,04

Fonte: Netcraft

**Tabela 2 - Total para usuários ativos através de todos os domínios**

Colaborador	Março 2006	Por cento	Abril 2006	Por cento	Mudança
Apache	24617922	67,20	24826483	64,88	- 2,32
Microsoft	9182372	25,07	9946789	25,99	0,92
Zeus	260185	0,71	257024	0,67	- 0,04
Sol	247825	0,68	236532	0,62	- 0,06

Fonte: Netcraft

No projeto, o Apache está sendo utilizado como o servidor web pelo seu baixo custo e facilidade na operação.

### 2.2.2 Navegador

Um navegador (também conhecido como web *browser* ou simplesmente *browser*) é um programa que habilita seus usuários a interagirem com documentos HTML hospedados em um servidor Web. É o tipo mais comumente usado de agente. A maior coleção interligada de documentos hipertexto, dos quais os documentos HTML são substanciais frações, é conhecida com a *World Wide Web*. [WIKIPEDIA,2006].



O *browser* utilizado no projeto é o *Firefox* por ser um navegador de código aberto, possuir inversões para muitos sistemas operacionais e boa portabilidade. A utilização desta ferramenta auxilia o desenvolvimento de softwares devido às funcionalidades particulares existentes em seu código, sendo assim o desenvolvedor torna-se mais hábil e eficaz.

## 2.3 BANCO DE DADOS

O banco de dados é um repositório, como o próprio nome diz: de dados, geralmente são agrupadas informações para uma mesma finalidade. Um depósito de informações que podem ser acessadas por usuários com a utilização de um programa chamado SGBD que é um gerenciador de banco de dados.

As informações podem ser acrescentadas e extraídas sem perda da fidelidade do dado. O modelo de dados mais adotado hoje é o relacional onde a estrutura tem forma de tabelas com linhas e colunas.

### 2.3.1 MySQL

O *MySQL* foi criado na Europa, mais especificamente na Suécia por uma equipe que trabalha junto desde a década de 1980. É um gerenciador de banco de dados que tem como interface a linguagem SQL. Atualmente, aproximadamente 70 profissionais no mundo inteiro estão empregados no desenvolvimento do sistema e mais de mil testam o software, fazem a integração com demais produtos e documentam as funcionalidades; fazendo assim com que este banco de dados seja um dos mais utilizados no mundo inteiro. [WIKIPEDIA,2003].

*MySQL* está ganhando o mercado de soluções de TI (Tecnologia da Informação) por suas vantagens, como por exemplo: fácil integração com softwares *Open Sources*, portabilidade para rodar em qualquer plataforma, bom desempenho e estabilidade, suporte a diversos tipos de tabela, compatibilidade com diversas interfaces e linguagens diferenciadas.

O uso do *MySQL* no projeto está sendo da seguinte maneira: todos os dados informados pelo usuário estão sendo gravados no banco e o sistema faz buscas de cinco em cinco segundos para verificar se ocorreu alguma alteração, interagindo assim com todas as interfaces.

## 2.4 LINGUAGENS DE PROGRAMAÇÃO PARA WEB

### 2.4.1 HTML

HTML significa Linguagem de Marcação de Hiper Texto (*Hypertext Markup Language*). É uma linguagem de marcação universal usada na criação de documentos para páginas na Internet. Criada por Tim Berners-Lee nos anos 90, ela está hoje em sua versão 4.0.1 e conta com inúmeras novidades e evoluções. [W3C, 1999]. Os códigos são interpretados pelos navegadores (*browser*).

Com a linguagem é permitida a formatação de textos, inserções de documentos de multimídia (imagens, vídeos e sons) através de marcações (*tags*) que possuem funções específicas e consistem em breves instruções tendo uma marca de início e outra de final: "<" e é fechada com o símbolo ">".

### 2.4.2 XHTML

O XHTML, ou *eXtensible Hypertext Markup Language*, é uma adaptação da linguagem HTML baseado em XML, combinando as *tags* do HTML com a padronização do XML aumentando assim a acessibilidade das páginas web por outros dispositivos que tenham um acesso ao navegador. Pode ser interpretado por diferentes plataformas, o que não era possível no HTML.

Desde 26 de Janeiro de 2000, se tornou a Recomendação oficial da W3C (*World Wide Web Consortium*) [W3C, 1999].

Por ter um formato mais rígido, com especificações e mais nível de detalhamento, adequa o uso da internet e melhora o desempenho e facilidade no acesso por um palm top, por exemplo.

### 2.4.3 CSS

HTML foi originalmente desenvolvida como uma linguagem estruturada, porém com a utilização foi se tornando uma ferramenta para suporte de fontes, cores, enfim, mais visual. *Cascading Style Sheets* oferece um diferencial na criação de páginas web com a intenção de retomar a definição original do HTML.

Assim, o CSS possui, de certa forma, um conceito inovador que possibilita a mudança da aparência simultânea de todas as páginas relacionadas com o mesmo estilo. O programador cria um *link* para uma página que contém os estilos,

procedendo de forma idêntica para todas as páginas de um portal. E para alterar a aparência do portal basta modificar apenas um arquivo correspondente. [WIKIPEDIA, 2006].

É utilizado no projeto para dar uma melhor “aparência” ao sistema, tornando-o mais amigável ao usuário final e consequentemente facilitando o manuseio do mesmo.

#### **2.4.4 AJAX**

AJAX (significa *Asynchronous Javascript And XML*) é o uso sistemático de *Javascript* e *XML* (e derivados) para tornar o navegador mais interativo com o usuário, utilizando-se de solicitações assíncronas de informações. AJAX não é somente um novo modelo, é também uma iniciativa na construção de aplicações web mais dinâmicas e criativas. Então, AJAX não é uma tecnologia, são realmente várias tecnologias trabalhando juntas, cada uma fazendo sua parte, oferecendo novos meios poderosos. [ADAPTIVE PATH,2006].

O AJAX roda no próprio navegador melhorando assim o desempenho das tarefas e procedimentos envolvidos. Para a execução é necessário algum navegador lançado após 2001.

Existem quatro princípios básicos para o AJAX: O navegador hospeda uma aplicação e não um conteúdo; o servidor fornece dados e não conteúdos; a interação do usuário com a aplicação pode ser flexível; e uma real codificação requer disciplina.

No sistema, o AJAX é explorado para dar flexibilidade de desenvolvimento proporcionando uma velocidade maior na troca de informação, evitando tempo de espera para carregar os dados. Sendo assim, no escopo do sistema, por ser de execução em tempo real, esta tecnologia permite essa funcionalidade.

#### **2.4.5 PHP**

PHP, *Hypertext Preprocessor*, é uma linguagem de programação de computadores interpretada, livre e muito utilizada para gerar conteúdo dinâmico na Web. Apesar de ser uma linguagem de fácil aprendizado e de uso para pequenos scripts dinâmicos simples, o PHP é uma linguagem poderosa orientada a objetos.

A linguagem surgiu por volta de 1994, como um subconjunto de scripts Perl criados por Rasmus Lerdorf. Com as adições de Zeev Suraski e Andi Gutmans, dois programadores israelitas pertencentes ao Technion, o Instituto Israelita de Tecnologia, que reescreveram o *parser*, era lançada em 1997 a PHP 3, primeira versão estável e parecida com a linguagem atual. Em maio de 2000 veio ao público a versão 4, e em julho de 2004, a versão 5.

Trata-se de uma linguagem extremamente modularizada, o que a torna ideal para instalação e uso em servidores web. É muito parecida, em tipos de dados, sintaxe e mesmo funções, com a linguagem C e com a C++. (A partir da versão 5, inclusive, PHP conta com um excelente suporte à orientação a objetos.) Pode ser, dependendo da configuração do servidor, embutida no código HTML. Além disso, destaca-se a extrema facilidade com que PHP lida com servidores de base de dados, como MySQL, PostgreSQL, Microsoft SQL Server e Oracle.

Existem versões do PHP disponíveis para diferentes tipos de sistemas operacionais. Em sua versão mais recente — o PHP5 — foi introduzido um novo modelo de objetos e, com isso, foram implementadas muitas funções e características firmando o PHP como linguagem Orientada a Objetos, e o colocando a novos níveis de concorrência com linguagens mais robustas, como o JAVA [ACHOUR, 2004].

A implementação do sistema por esta linguagem é feita pela manipulação das informações que fazem a interação do usuário com a base de dados. É também o meio de comunicação entre toda a estrutura física do projeto, incluindo: servidor, interface e saída.

## 2.5 COMPONENTES FÍSICOS

### 2.5.1 Computador Pessoal

A expressão "computador pessoal" é geralmente abreviada para a sigla PC, do inglês "*personal computer*". Também é comum a referência *desktop* a esta classe de computador. [WIKIPEDIA, 2006].

Na implementação física do projeto, o PC é utilizado no módulo do Caixa e no da Cozinha, sendo que em ambos os casos têm-se atividades de leitura e escrita no sistema.

### 2.5.2 Porta Paralela

A interface paralela é um dos pontos de onde o computador pode se comunicar com o mundo externo, ou seja, uma interface de entrada e saída (E/S ou I/O do inglês). Esta porta, tradicionalmente unidirecional, evoluiu ao longo do tempo, permitindo uma comunicação bi-direcional e mantendo sua proposta básica de transmissão paralela. Isto implica na transmissão byte a byte, pois esta porta transmite paralelamente 8 bits de uma só vez, levando a um número maior de fios e uma capacidade de distância de transmissão curta (cerca de oito metros), devido à interferência de ruídos. [TORRES, 2002a]

A comunicação do display utilizado para interagir com o sistema na troca de informações é feita pela porta paralela devido à sua facilidade e maiores possibilidades de endereçamento. Permitindo assim, uma maior quantidade de dados enviados e recebidos.

### 2.5.3 Ponto-de-Acesso

Numa rede *wireless*, o *hub* é substituído pelo ponto de acesso (*access-point* em inglês), que tem a mesma função central que o *hub* desempenha nas redes com fios: retransmitir os pacotes de dados, de forma que todos os micros da rede os recebam.

Os pontos de acesso possuem uma saída para serem conectados num *hub* tradicional, permitindo que você "junte" os micros da rede com fios com os que estão acessando através da rede *wireless*, formando uma única rede, o que é justamente a configuração mais comum.

Existem poucas vantagens em utilizar uma rede *wireless* para interligar micros *desktops*, que afinal não precisam sair do lugar. O mais comum é utilizar uma rede cabeada normal para os *desktops* e utilizar uma rede *wireless* complementar para os *notebooks*, *palmtops* e outros dispositivos móveis.

É utilizado um *hub/switch* tradicional para a parte cabeada, usando cabo também para interligar o ponto de acesso à rede. O ponto de acesso serve apenas como a "última milha", levando o sinal da rede até os micros com placas *Wireless*. Eles podem acessar os recursos da rede normalmente, acessar arquivos compartilhados, imprimir, acessar a internet, etc. A única limitação fica sendo a

velocidade mais baixa e o tempo de acesso mais alto das redes *wireless*. [GUIA DO HARDWARE,2006].

O projeto utilizará um ponto de acesso no servidor permitindo que os demais componentes estejam conectados na rede do estabelecimento. O ponto de acesso diminuirá a quantidade de componentes físicos (cabeamento USB) e possibilitará um melhor layout estrutural da rede.

#### **2.5.4 Display**

Um *display* de cristal líquido, ou LCD, é um monitor muito leve e fino sem partes móveis. Consiste de um líquido polarizador da luz, eletricamente controlado que se encontra comprimido dentro de celas entre duas lâminas transparentes polarizadoras. Os eixos polarizadores das duas lâminas estão alinhados perpendicularmente entre si. Cada cela é provida de contatos elétricos que permitem que um campo elétrico possa ser aplicado ao líquido no interior. A tecnologia é utilizada há tempos em calculadoras, relógios digitais, entre outros acessórios. [WIKIPEDIA, 2007].

No projeto, o *display* será utilizado para mostrar ao cozinheiro qual o pedido ele precisa cozinhar. Foi escolhido um *display* com dezesseis caracteres por duas linhas, algo pequeno que não seria danificado na cozinha devido à sujeira e gordura.

#### **2.5.5 PDA**

*Personal Digital Assistants* (PDAs ou *Handhelds*), ou Assistente Pessoal Digital, é um computador de dimensões reduzidas (cerca de 11cm x 7 cm x 1,5cm), dotado de grande capacidade computacional, cumprindo as funções de agenda e sistema informático de escritório elementar, com possibilidade de interconexão com um computador pessoal e uma rede informática sem fios para acesso a correio eletrônico e internet. [WIKIPEDIA, 2006]

Os PDAs estão cada vez mais sofisticados, com capacidade maior de armazenamento, softwares diversos, tela colorida, acesso à internet, câmera digital, MP3, rede *wireless*, *bluetooth*, e demais tecnologias que encontramos no mercado, aproximando-se cada vez mais dos computadores de mesa.

As principais famílias encontradas são:

- *PalmOne*: utilizam o sistema operacional *Palm OS* da *PalmSource* (ex-*Palm Computing*), um sistema rápido e bastante confiável, que, mesmo restrito quanto ao número de fabricantes que o adotaram, é atualmente o mais utilizado no mundo.
- *Pocket PC*: utilizam o sistema *Windows Mobile* (também conhecido como *Pocket PC* e anteriormente chamado como *Windows CE*) da Microsoft, que é "compatível" com o Windows e foi adotado por uma gama bem variada de fabricantes de PDA. Hoje em dia a última versão do sistema operacional para *Pocket PC* é a *Windows Mobile 5.0*. Estes PDAs utilizam programas que possibilitam a troca de dados com programas do Windows dos computadores de mesa. Um programa feito para *Pocket PC* não roda no sistema operacional Windows (95,98,ME, XP) do *desktop*, e vice-versa. Apesar de ambos serem da Microsoft são sistemas totalmente diferentes.

O PDA será utilizado no projeto como o “cardápio eletrônico” para o cliente. Ou seja, pelo PDA o cliente poderá ver os produtos oferecidos, fazer o seu pedido, ver a conta parcial e solicitar o encerramento da mesma.

## **CAPÍTULO 3 – DESENVOLVIMENTO**

Este capítulo tem como objetivo apresentar a análise desde a proposta em desenvolver um sistema inteligente que integrasse diversas tecnologias até a implementação final do projeto, que recebe o nome de SisMannu.

Será apresentado em quatro partes: conceitos iniciais, custos, módulos e interações entre as tecnologias.

### **3.1 CONCEITOS INICIAIS**

O SisMannu é um sistema projetado para permitir o gerenciamento de pedidos realizados por clientes em um restaurante. A partir de um PDA, localizado na mesa do cliente, o mesmo poderá solicitar seus pedidos e acompanhar o valor de sua conta. Aliado a isso o responsável pelo estabelecimento saberá todas as ocorrências de pedidos através de qualquer computador em sua rede local.

Este aplicativo foi desenvolvido com o intuito de fazer com que os clientes de um restaurante possam realizar os seus pedidos de maneira prática e ágil. Contudo, seguindo a mesma idéia de praticidade e agilidade, o sistema possui três módulos WEB, sendo um deles acessado pelo PDA, e dois módulos acessados via PC, para o gerenciamento das mesas e dos pedidos realizados ao estabelecimento, tornando-o assim independente de sistema operacional.

Cada mesa do estabelecimento estará registrada no sistema, pois através dela, todos os pedidos serão controlados pelo responsável da cozinha, que será chamado de Balcão, para que os mesmos possam ser entregues com prontidão ao cliente, de acordo com a ordem de chegada. Com isso o cliente, terá a certeza da entrega de seu pedido de acordo com o momento de sua solicitação.

As solicitações poderão ser vistas pelo responsável do caixa, que será chamado de Caixa, para que possa verificar o andamento das contas, clientes e gerenciar os garçons responsáveis por cada mesa.

#### **3.1.1 Interações com os usuários**

O sistema possui três módulos:

- Módulo Cliente;



- Módulo Balcão, e;
- Módulo Caixa.

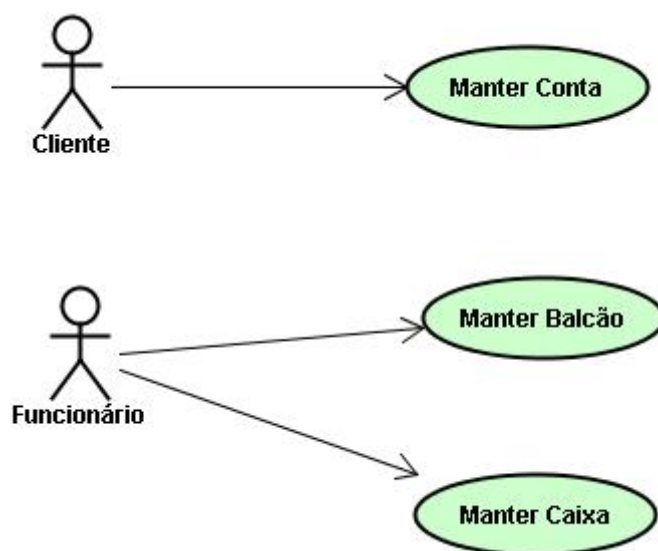


Figura 2 - Diagrama de Caso de Uso

Com base na **Figura 2**, pode-se observar que o Funcionário possui dois tipos de interação com o sistema. Sendo elas separadas por perfis de acesso, Balcão e Caixa. Assim, resta apenas o gerenciamento da conta por parte do Cliente.

O caso de uso Manter Conta tem início quando o cliente é orientado a realizar o login, conforme **Figura 3**. Em seguida o mesmo poderá escolher o seu pedido através de uma lista de produtos e realizar o seu pedido. Após a realização do pedido a conta poderá ser solicitada pelo cliente. As figuras **Figura 4** e **Figura 5** exemplificam estes casos.

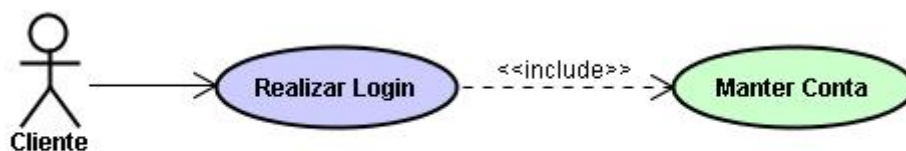


Figura 3 - Caso de Uso: Realizar Login

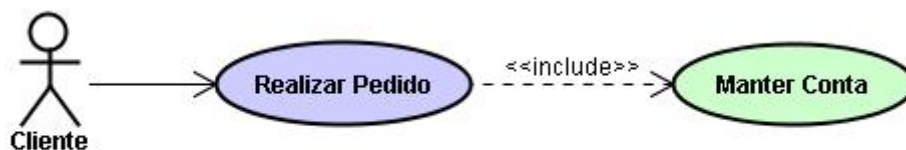


Figura 4 - Caso de Uso: Realizar Pedido

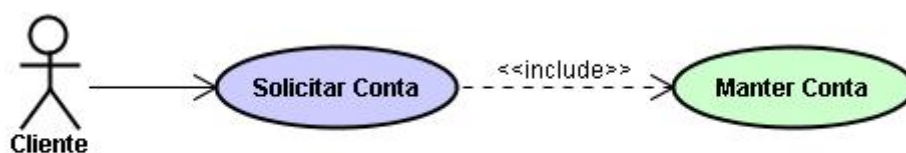


Figura 5 - Caso de Uso: Solicitar Conta

O caso de uso Manter Balcão se inicia quando o Funcionário realiza o login no sistema (**Figura 6**), sendo que para acessar este módulo o usuário deverá possuir o perfil de Balcão. Este perfil permite a visualização (**Figura 7**) e liberação de todos os pedidos (**Figura 8**) solicitados ao estabelecimento, separados por mesas e ordem de chegada.

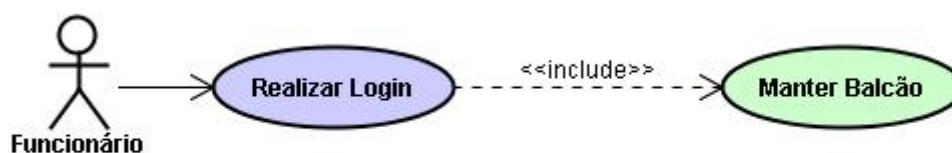


Figura 6 - Caso de Uso: Realizar Login (Perfil Balcão)

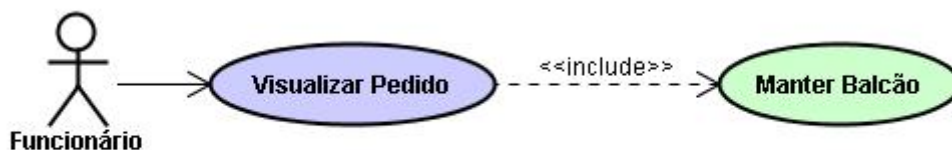


Figura 7 - Caso de Uso: Visualizar Pedido

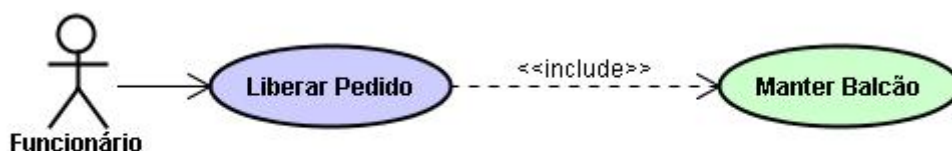


Figura 8 - Caso de Uso: Liberar Pedido

O caso de uso Manter Caixa se inicia quando o Funcionário realiza o login no sistema (**Figura 9**), sendo que para acessar este módulo o usuário deverá possuir o perfil de Caixa. Neste módulo o funcionário poderá alocar os garçons responsáveis por cada mesa (**Figura 10**) a qualquer momento. Em cada item que representa a mesa, o funcionário poderá visualizar as contas ativas, verificar o histórico das contas do dia (**Figura 11**), e após solicitação do cliente para o término da conta, o funcionário encerrará e liberará a mesa para o próximo cliente (**Figura 12**).

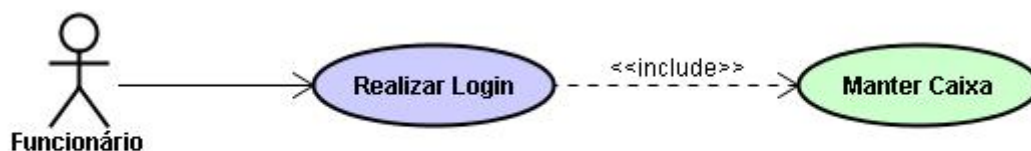


Figura 9 - Caso de Uso: Realizar Login (Perfil Caixa)



Figura 10 - Caso de Uso: Gerenciar Garçom



Figura 11 - Caso de Uso: Gerenciar Conta

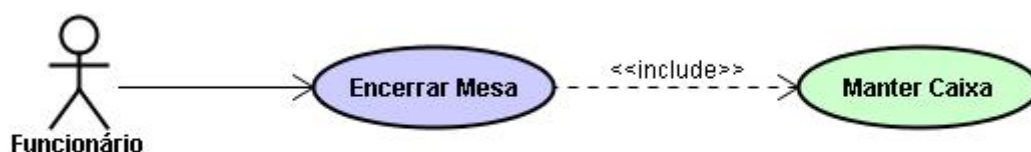


Figura 12 - Caso de Uso: Encerrar Mesa

### 3.1.2 Modelagem do Sistema

O sistema foi desenvolvido utilizando o conceito de módulos, onde cada um possui sua atribuição de forma bem definida. Apesar de utilizar este conceito, um módulo depende da ação de outro para que aconteça a total interação do negócio. Sua arquitetura é totalmente baseada em sistemas web utilizando PHP, Javascript e MySQL, seja ela para PDA ou desktop. Dessa forma, podemos separar o sistema em três módulos:

- Módulo PDA

Este módulo consiste em uma aplicação WEB voltada para dispositivos móveis. Seu acesso é permitido pela utilização do navegador web instalado no PDA, dando assim a possibilidade ao cliente para que acesse o cardápio e realize os seus

pedidos. Por este módulo também é possível verificar a conta parcial e solicitar o fechamento da conta.

- Módulo Balcão

Módulo acessado através de um desktop localizado no balcão, permitindo ao funcionário do estabelecimento, verificar os pedidos solicitados pelos clientes e após a entrega desses pedidos, realizar o fechamento do mesmo.

- Módulo Caixa

É o módulo de controle administrativo do sistema. Este permite a visualização de todas as mesas e seus respectivos pedidos, bem como o histórico das contas do dia atual. Através deste módulo, o funcionário poderá fechar uma conta e liberar a mesa para um novo acesso.

A interação entre os três módulos pode ser observada na **Figura 13**, que mostra o diagrama de arquitetura do SisMannu.

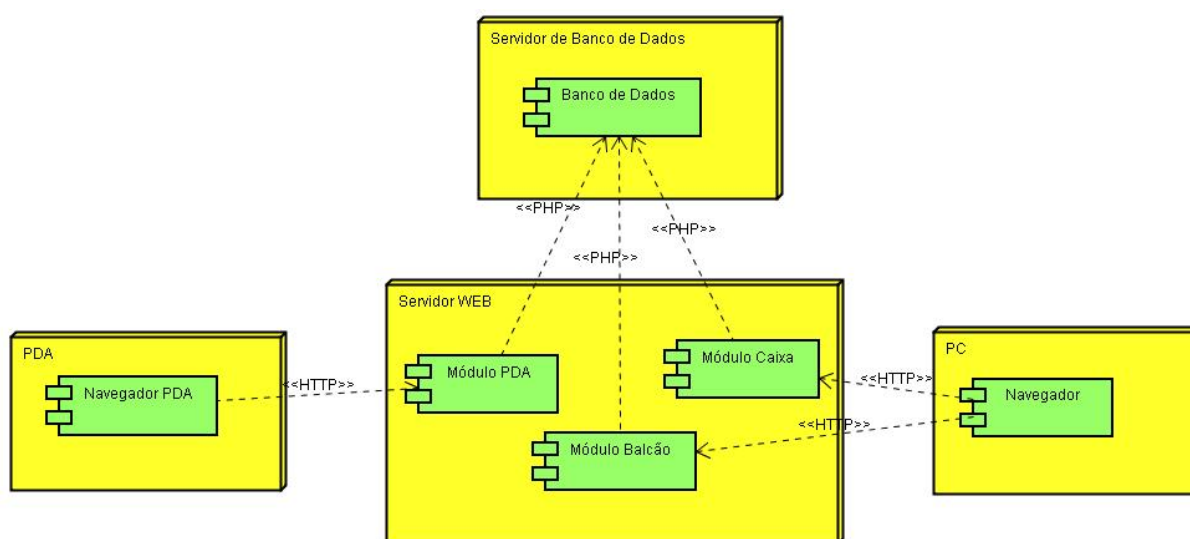


Figura 13 - Diagrama de Arquitetura

Os módulos encontram-se em uma máquina composta por um servidor WEB e um servidor de banco de dados. O servidor WEB através da linguagem PHP é responsável por encaminhar as solicitações do PDA e do PC até o banco de dados, permitindo assim as possíveis atualizações.

Esta máquina também é responsável pela comunicação física entre todos os dispositivos, sendo esta comunicação via cabo USB ou por uma rede local.

É importante salientar que o navegador do PDA deve possuir um navegador WEB com suporte a CSS (*Cascading Style Sheet*, ou seja folhas de estilos em cascata), uma vez que toda a sua interface é montada utilizando esta linguagem.

### 3.1.3 Modelagem do Banco de dados

O banco de dados foi desenvolvido de acordo com a regra de negócio do sistema definido nos conceitos iniciais e pelos objetos definidos na modelagem do sistema.

Ao longo do desenvolvimento, algumas mudanças foram feitas devido à dificuldade da compreensão de todos os componentes envolvidos ou para poder incrementar uma regra pré-definida.

As informações constantes, contidas nas tabelas: mesa, garçom, usuário e produto são armazenadas no banco na implantação do sistema. E as informações consideradas variáveis, ou seja, que sofrem atualizações em tempo de execução estão contidas nas tabelas: cliente, pedido e conta.

O MySQL, versão 4, foi escolhido por motivos de baixo custo e facilidade na interação com a linguagem de programação PHP. Como o sistema possui diversos módulos e interfaces, atingir um baixo custo acaba sendo um fator importante para a escolha das ferramentas a serem utilizadas. O MySQL também possui um bom processamento, boa performance na troca de informações com o sistema.

Como pode ser verificado no Modelo de Entidade-Relacionamento (**Figura 14**), um cliente pode estar associado a várias contas, pois cada vez que retorna ao estabelecimento, uma nova conta é criada no banco. Uma conta contém vários pedidos, considerando que pedido é a ação que o cliente faz no sistema de enviar por vez os determinados produtos que gostaria e suas respectivas quantidades, então determina-se também que um pedido possui diversos produtos. Uma mesa contém várias contas sucessivas, nunca podendo ter duas contas em aberto. E para cada mesa pode ser definido um garçom. (Ver Apêndice II)

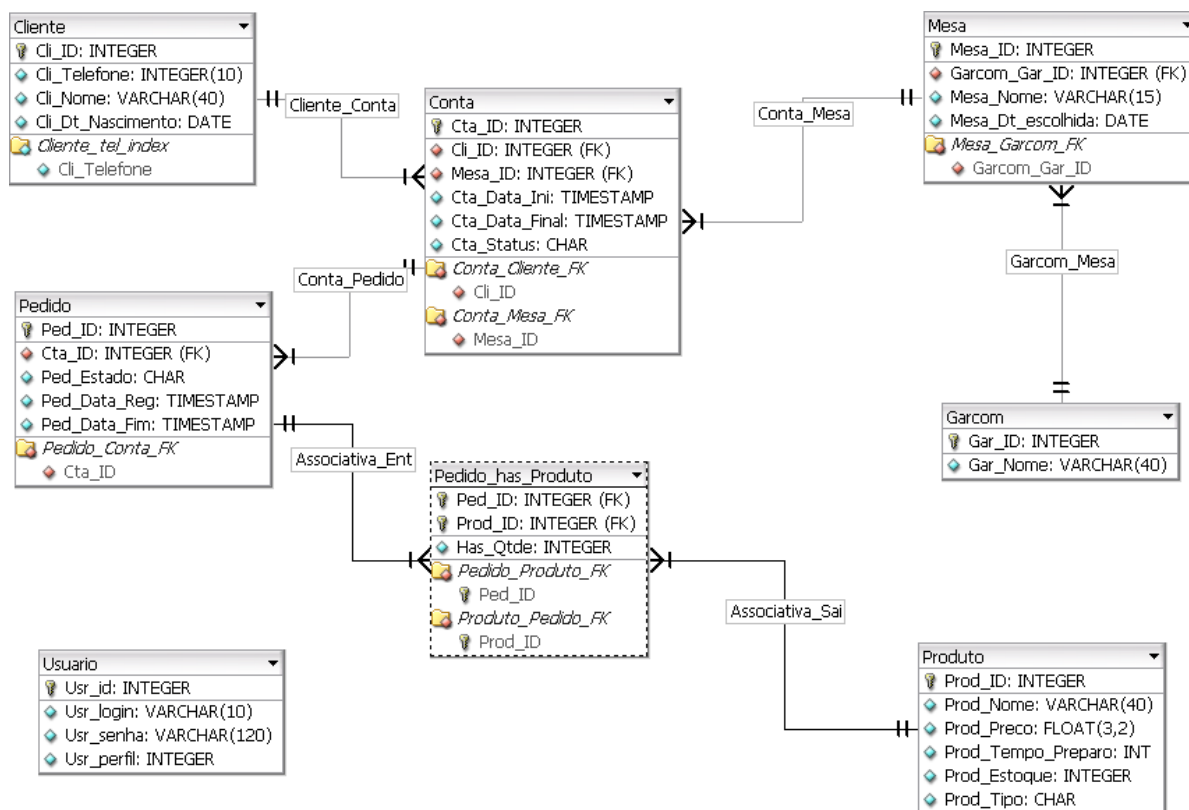


Figura 14 - Modelo de Entidade-Relacionamento

### 3.1.4 Implementação do Sistema

No desenvolvimento do projeto, houve uma preocupação em manter organizado todo o ambiente em que o sistema será executado. Assim, existe uma separação lógica e estrutural nos diretórios e pastas, tanto de configurações do ambiente (PHP, MySQL e Apache) quanto do sistema propriamente dito.

O diretório principal que contém todo o sistema foi chamado de AppServ (Figura 15). Este diretório está dividido em duas pastas: Apache Group e www.



Figura 15 – Diretório Principal

Na pasta Apache Group, estão os arquivos configurados do Apache, do MySQL e do PHP (Figura 16), sendo que o motivo de estarem as três ferramentas no mesmo diretório é somente para obter uma melhor mobilidade quando quiser

mudar o ambiente em que será executado, evitando assim que alguma informação se perca.

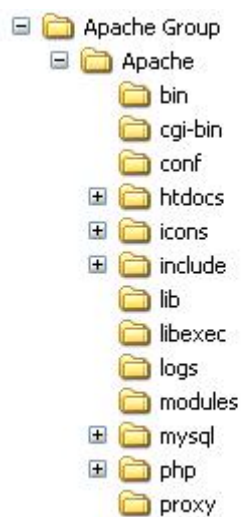


Figura 16 – Estrutura da pasta Apache Group

Na pasta www, está efetivamente o sistema e suas sub-pastas (**Figura 17**), onde cada classe, biblioteca, imagens, módulos e templates estão separados para obter uma maior facilidade na implementação e na manutenção.



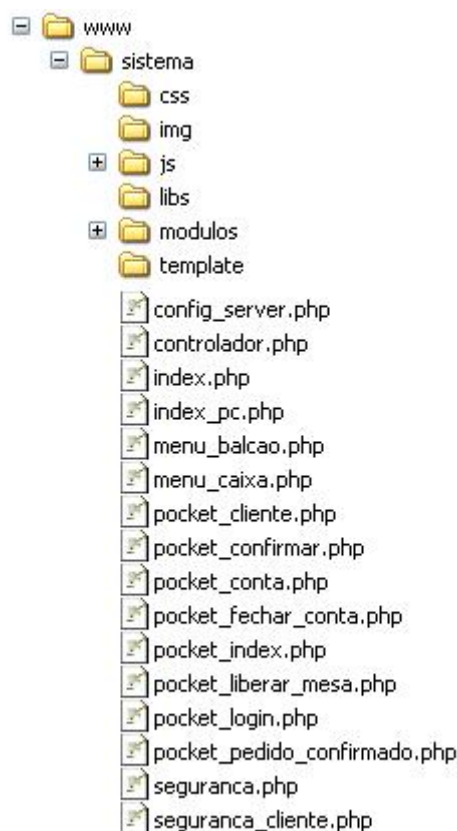


Figura 17 – Estrutura da pasta www

As definições de cada pasta são:

1. CSS: estão os arquivos com os estilos das páginas, que definem a formatação de texto (cor, fonte, tamanho), cores de linhas e tabelas, entre outros.
2. IMG: possui todas as imagens utilizadas no sistema para cada módulo.
3. JS: estão os arquivos em *JavaScript*, que são responsáveis pela comunicação assíncrona (Ajax).
4. LIBS: estão todas as classes utilizadas para auxiliar o desenvolvimento e implementação do sistema. Nele estão disponíveis as classes de comunicação com o banco de dados, classes de negócio (**Figura 18**) e classes utilitárias com funções personalizadas, bem como um arquivo centralizador contendo todos estes arquivos, que é o *Fachada.php*.

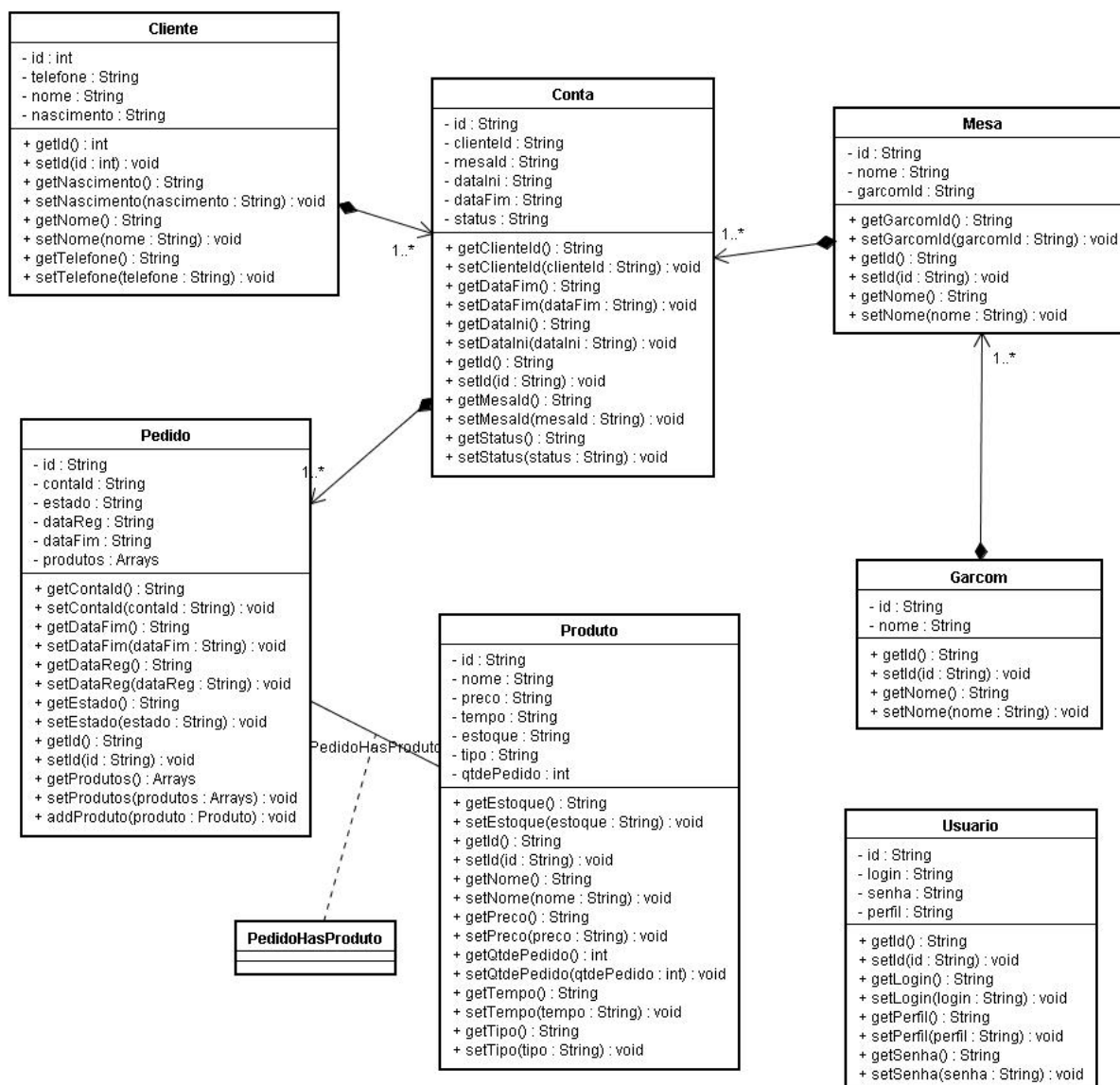


Figura 18 - Diagrama de classes

5. **MODULOS:** encontram-se os arquivos que serão chamados de forma assíncrona ou via PDA, de acordo com o seu módulo. São os responsáveis em buscar as classes utilitárias para apresentar a interface ao usuário.
6. **TEMPLATE:** possui o cabeçalho e o rodapé das páginas.
7. **RAIZ:** estão os arquivos principais de cada módulo, os de controle, os de chamada ao PDA e os de segurança.

### 3.2 CUSTO PARA IMPLEMENTAÇÃO

Seguindo um dos objetivos do projeto que é a redução de custos, foram adotados os seguintes propósitos para a implementação:

- Redução no valor do software;
- Tecnologia avançada para o PDA;
- Hardware simples com bom funcionamento.

Foram utilizados softwares chamados *Open Source* e softwares que não possuem custo, como Java e PHP. O banco MySQL e o Apache também não possuem custo. Como o projeto tem um intuito puramente acadêmico, o custo de implantação e manutenção não está sendo considerado, entretanto, caso a comercialização fosse feita, teria um custo para estas duas fases.

Para a utilização do PDA, é preciso que o *palm* suporte um navegador e que permita a conexão ou por cabo USB ou conexão wireless, ou seja, permita que o dispositivo consiga estar conectado à rede local do estabelecimento. Foi comprado um palm por R\$2.000,00 com a seguinte configuração:

- HP IPAQ rx 3715;
- Processador Samsung S3C2440;
- Memória 64Mb RAM e a 128Mb ROM;
- Wireless;
- Windows Mobile 2003 SE com o Microsoft Office;
- Internet Explorer.

Na escolha do display LCD, primeiramente foi comprado uma porta paralela por R\$9,00 e um display por R\$30,00. Em seguida foi adquirido um Kit com os dois itens já conectados por R\$40,00, sendo um custo pequeno com um dispositivo de boa configuração para atender o projeto.

Portanto, o custo total do projeto é acessível aos estabelecimentos que queiram implementá-lo, sendo que o custo do PDA pode ser reduzido uma vez que existam séries mais novas e com menor preço.

### 3.3 MÓDULOS

#### 3.3.1 Caixa

O módulo **Caixa** é considerado o módulo de Administração porque o usuário que o opera é responsável por:

- Determinar o garçom para cada mesa;
- Acompanhar a conta parcial de cada cliente;
- Emitir conta total de cada cliente;
- Encerrar conta em aberto;
- Liberar a mesa para um novo cliente.

A tela de início, chamada Tela de Login (**Figura 19**), é comum ao módulo **Caixa** e ao módulo **Balcão**.

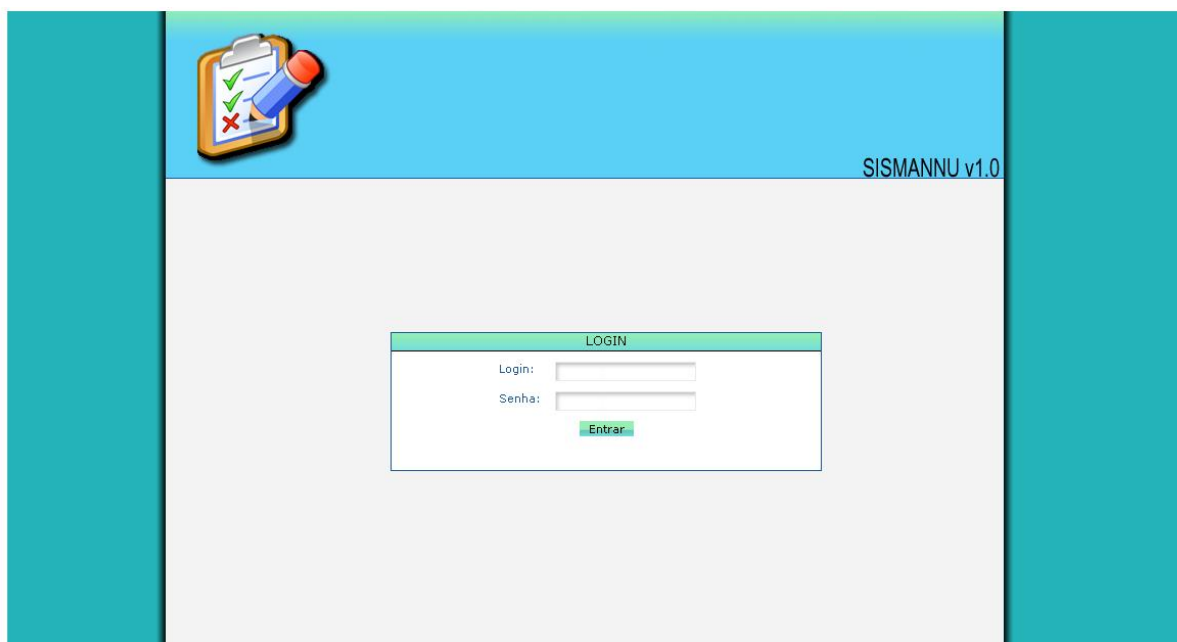


Figura 19 – Tela de Login

Existem dois usuários pré-cadastrados na tabela Usuário que determinam qual perfil de acesso será acionado para operar o sistema, no caso do módulo **Caixa**, o login e o perfil também são chamados de *caixa*. Na tela de login, Usuário e Senha serão validados, e caso alguma informação esteja errada, um comunicado de *senha/login inválidos* aparecerá.

Ao entrar no sistema, o usuário Caixa verá a Tela Principal do sistema (**Figura 20**). A tela principal no caixa indica o status de cada mesa com seus respectivos garçons, clientes, pedidos, produtos e contas. Para implementação em banca, foi considerado que o local tivesse apenas quatro mesas, porém seria possível implementar para uma quantidade maior também.

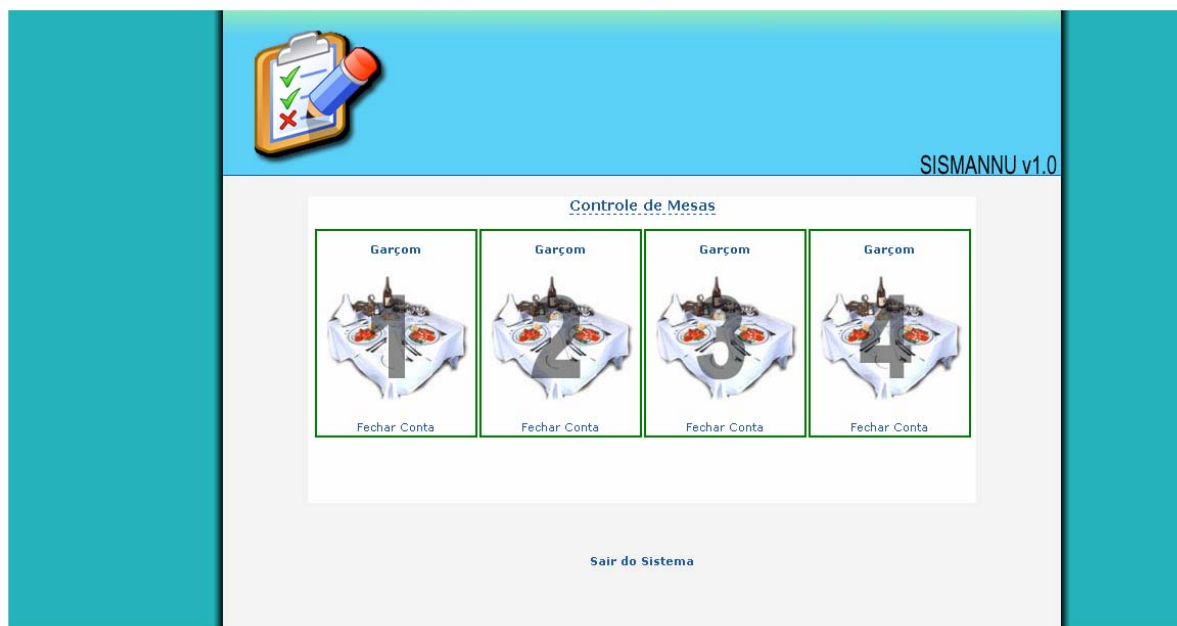


Figura 20 – Tela Principal - Caixa

Para facilitar o entendimento e aproximar do conhecimento comum que temos na sociedade, foi escolhido o padrão adotado de cores pelos semáforos para sinalizar as ações que ocorrem em uma mesa. Ou seja, para demonstrar visualmente que uma ação foi realizada, foram adotadas as cores: verde, amarelo e vermelho, com seus respectivos significados para a logística de tráfego: disponível, atenção e fechado.

Assim, no desenvolvimento, foi adotado que a cor verde ao redor da imagem da mesa representaria o status *aberto*, representando que está desocupada, ou seja, sem cliente. A partir do momento que um cliente chega ao estabelecimento e senta em uma mesa, o status da mesma será alterado para *ocupado*, representado pela cor vermelha (**Figura 21**).



Figura 21 – Mesa Ocupada

Com a mesa ocupada ou desocupada, o Caixa também tem a função de coordenar os garçons responsáveis por cada mesa e o remanejamento pode ser feito em qualquer momento. Essa funcionalidade foi implementada para melhorar a distribuição do trabalho de entrega no local e para que um garçom não fique sobrecarregado enquanto os outros não são solicitados. É considerada como uma funcionalidade propriamente de gerência, sendo restrita somente ao administrador (Caixa). Para executar essa tarefa, foi criado um link na parte superior de cada imagem de mesa denominado Garçom, e ao clicar, uma caixa de seleção será aberta para a escolha do funcionário. Após a seleção, é necessário o clique em Alterar para que a tabela Garçom do banco seja atualizada.



Figura 22 – Coordena Garçom

Uma possibilidade que o Caixa tem no sistema é acompanhar a conta parcial e total de cada cliente. Após o cliente ter consumido o seu pedido, ele fará a solicitação do encerramento da conta e somente o perfil caixa possui esse privilégio. Quando o cliente executa a atividade *Encerrar Conta*, o status será alterado na Tabela Conta e a tela principal do caixa piscará uma cor amarela, demonstrando a última simbologia adotada que faltava ser comentada. O intuito foi justamente chamar a atenção do Caixa de que uma mesa está requisitando uma atividade que depende somente dele, pois quando o indicador está verde ou vermelho é só uma sinalização de que um status foi alterado, entretanto com o indicador amarelo, o Caixa precisa realizar uma tarefa que é a *Fechar Conta*.



Figura 23 – Encerrar Conta

Ao selecionar a mesa que pediu o Encerramento da Conta, a Tela de Controle (Figura 24) aparecerá e o Caixa precisa Fechar a Conta para que nenhum pedido seja feito a partir daquele momento, sendo assim, o sistema fica bloqueado para o terminal do cliente.



Figura 24 – Tela de Controle



Na mesma tela ainda, após o pagamento efetivado pelo cliente, o caixa tem a função de *Liberar Mesa* para que o sistema reinicialize o ambiente para um novo cliente. Ao lado direito da tela, existe um histórico das contas anteriores que o caixa pode abrir para fins de gerência, fechamento de caixa, balanço, entre outras finalidades do dia. E ao fechar a Tela de Controle, o sistema retornará à Tela Principal.

O fluxo de funcionamento para este módulo pode ser entendido também pela **Figura 25**.

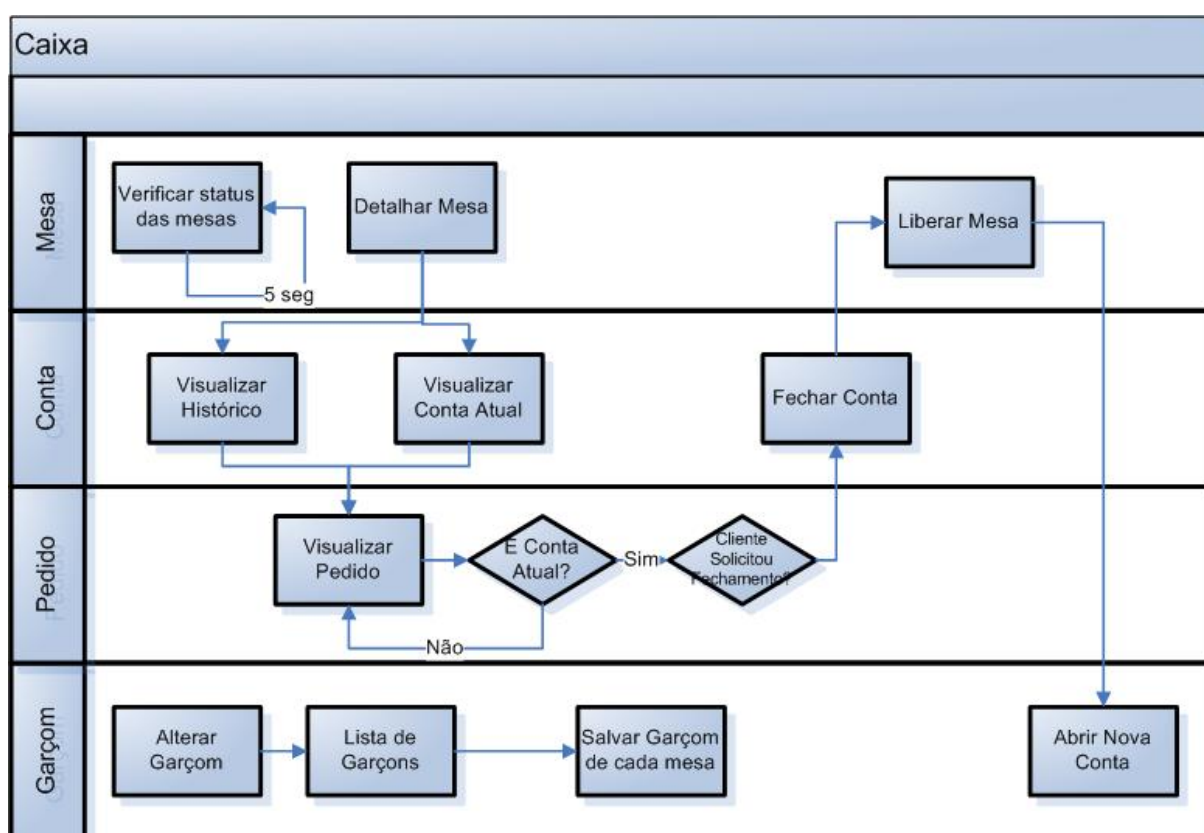


Figura 25 – Fluxograma do Caixa

### 3.3.2 Balcão

O módulo **Balcão** tem como finalidade principal *Liberar Pedidos*. Em outras palavras, a função do garçom no estabelecimento é verificar se o pedido está sendo feito e em seguida, entregá-lo.

Conforme colocado anteriormente, a Tela de Login (**Figura 19**) é comum ao módulo **Caixa** e módulo **Balcão**. Sendo que para este módulo, foi criado na tabela Usuário, um login e uma senha para acessar este perfil, chamado de *balcao*.

A Tela Principal do módulo Balcão também foi desenvolvida a partir do modelo da tela do Caixa, até mesmo para manter uma integridade maior do sistema. A exceção está apenas na funcionalidade Garçom, que é definida somente para o Caixa.

Ao ser realizado um pedido pelo cliente, a tela principal do Balcão altera seu status para vermelho, indicando que um novo pedido foi feito e precisa ser executado.

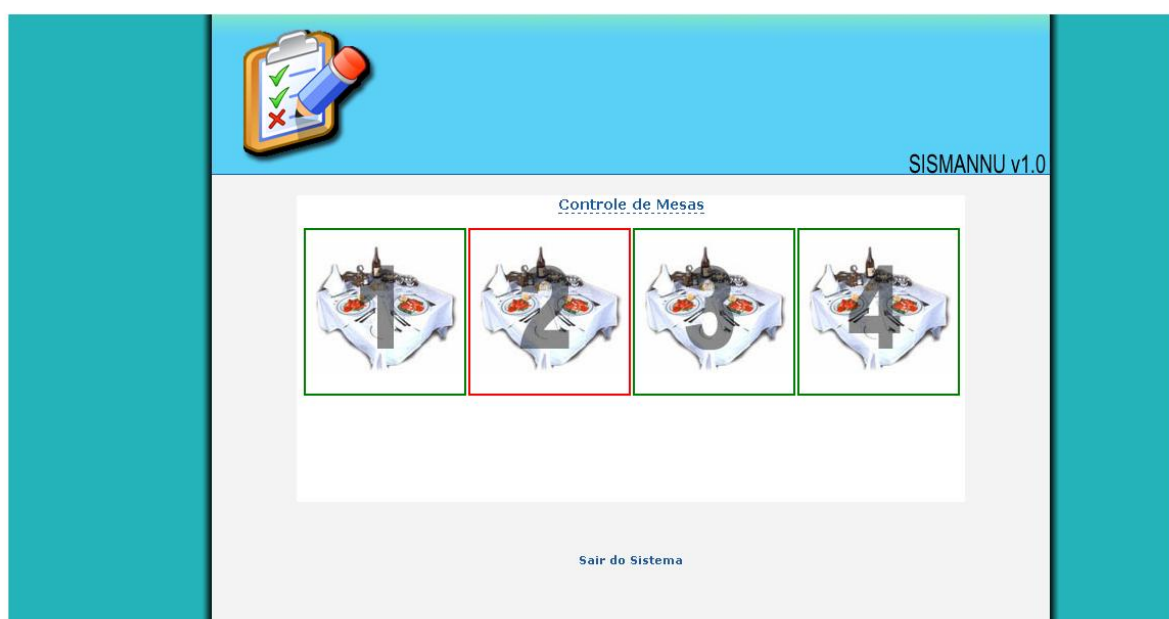


Figura 26 – Novo Pedido

Existe na modelagem dos dados um tipo para cada produto que foi classificado em *Bebida* e *Comida*. Essa diferença foi desenvolvida para ter uma melhor logística no estabelecimento. Assim sendo, quando o pedido for uma Bebida, o funcionário que estiver no Balcão é o responsável por pegar e disponibilizar para o Garçom entregar. Quando o pedido for uma comida, o display LCD que estará na cozinha mostrará no leitor ótico qual comida precisa ser feita e o cozinheiro será o encarregado de prepará-la. O display mostrará sempre os primeiros seis pedidos da fila, sendo respeitado o objetivo de garantir ao cliente que seu pedido será atendido conforme ordem de chegada. Adiante será explicado o funcionamento do hardware.

Para consultar os pedidos que o cliente fez, o responsável entrará na Tela de Controle de Pedido (**Figura 27**), clicando em cima da mesa indicada. Quando o

pedido estiver pronto para ser entregue, clicará na tarefa Liberar Pedido, disponibilizando os produtos ao Garçon que os entregará em seguida. Uma finalidade do *Liberar Pedido* é liberar também a fila de pedidos para as demais mesas receberem os seus itens escolhidos.



Figura 27 - Tela de Controle de Pedido

O fluxo de funcionamento para este módulo pode ser entendido também pela **Figura 28**.

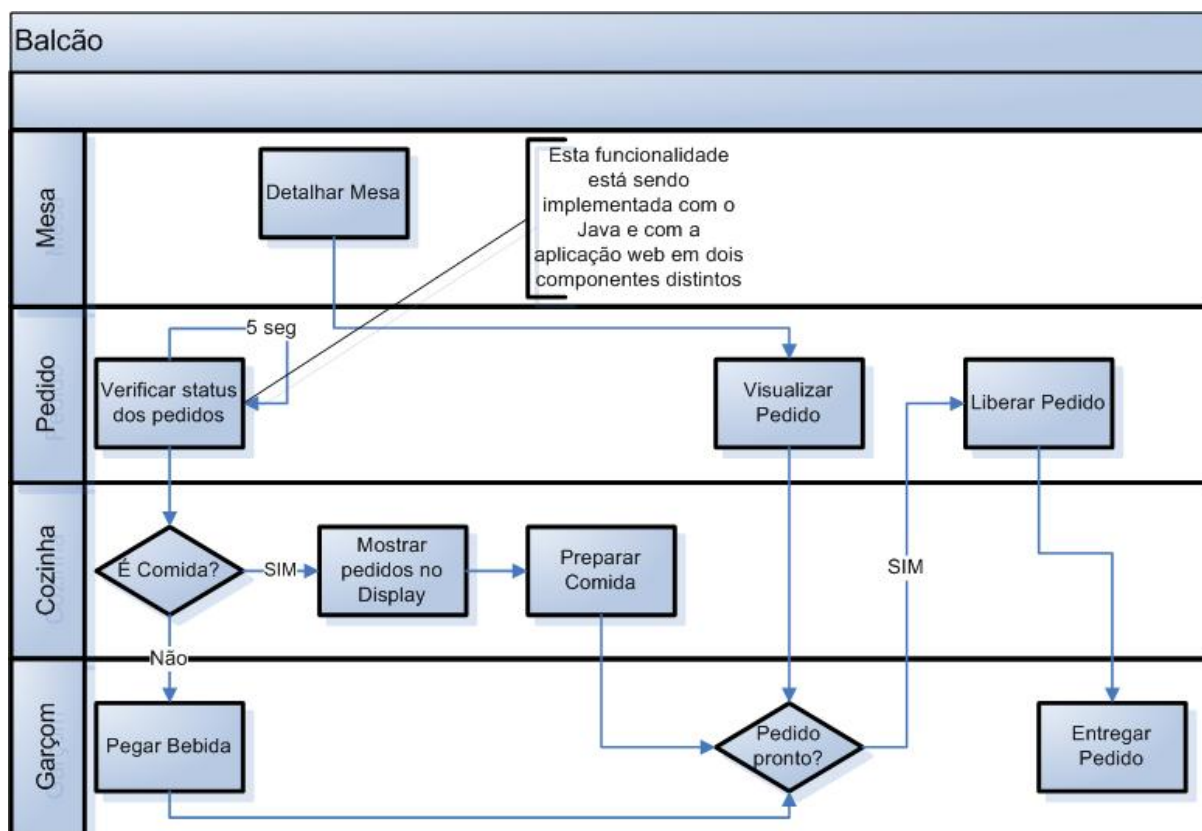


Figura 28 – Fluxograma do Balcão

### 3.3.2.1 COZINHA

A cozinha é colocada no projeto como uma extensão do Balcão, porém sem ser um módulo exclusivo no sistema. O pedido que chega ao balcão é o mesmo que precisa ser feito pela cozinha, segundo a ordem de chegada. Portanto, foi escolhido um dispositivo pequeno, evitando que fosse colocado um monitor maior o qual seria prejudicado devido à gordura, o calor, e outros fatores danosos, encontrados numa cozinha.

Para implementação, foi escolhido um display LCD para mostrar ao cozinheiro qual o produto ele precisa preparar. No display, aparecerão sempre os seis primeiros produtos selecionados, respeitando a ordem de execução dos pedidos.

No desenvolvimento do projeto, foi escolhida a linguagem de programação Java para realizar a comunicação macro entre o banco de dados e o display, devido à sua facilidade de execução ao iniciar o sistema operacional, por realizar suas atividades independentemente de ter um servidor web (desde que o banco de dados esteja ativo), e para poder utilizar mais uma ferramenta, referenciando ao objetivo do

projeto que é a sinergia entre diversas tecnologias. A integração entre os módulos está no item **3.3** Erro! Fonte de referência não encontrada..

Conforme descrito anteriormente no item Balcão, somente o produto que for do status *comida* será apresentado no display. Para o funcionamento no estabelecimento, quando o produto estiver pronto, o cozinheiro deverá colocá-lo à disposição do balcão para a entrega, e neste momento o responsável pelo balcão acionará a tarefa *Liberar Pedido* no sistema e a fila de prioridade será reorganizada. O tempo de cozimento e preparação dos alimentos foi desconsiderado pois estava fora do escopo do projeto acadêmico.

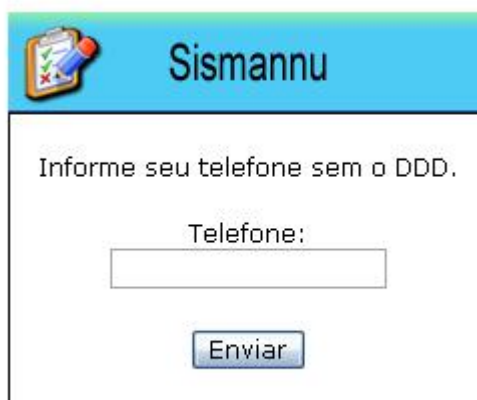
### 3.3.3 Cliente

O módulo **Cliente** é o responsável pela comunicação “amigável” com o cliente do estabelecimento. Foi desenvolvido no mesmo layout que os demais, porém com dimensões menores, apropriadas ao tamanho do PDA. A idéia do módulo era aproximar ao máximo de um cardápio eletrônico de fácil navegação entre as páginas.

Antes de iniciar o expediente, o garçom deverá ir a cada mesa e definir logisticamente no PDA qual é o número associado à ela.

Para o funcionamento, o passo inicial é o cliente chegar ao estabelecimento e escolher uma mesa disponível. Ao sentar-se, o Garçom fará a recepção e explicará o funcionamento básico do sistema, caso o cliente seja uma pessoa que não goste de “lidar com máquinas”. Para os clientes, inclusos digitalmente, o manuseio será fácil, visto que é um dos objetivos iniciais.

Na Tela de Login do Cliente (**Figura 29**) é solicitado o telefone do cliente. O telefone foi considerado como uma chave comum, evitando que fossem fornecidos códigos pessoais que trariam o esquecimento ou desconforto para o cliente. Portanto, foi escolhido o telefone como um número fácil de lembrar que ficaria associado ao cadastro de cada um.



**Sismannu**

Informe seu telefone sem o DDD.

Telefone:

Enviar

Figura 29 – Tela de Login do Cliente

Ao informar o telefone e enviar a informação, será aberta a Tela de Cadastro (**Figura 30**). Se o cliente estiver pela primeira vez no recinto, ele deverá informar o nome e a data de nascimento. Caso seja um cliente antigo, seus dados serão recuperados no banco e precisarão apenas da confirmação. A finalidade de obter essas informações do cliente é puramente comercial, visando que o estabelecimento queira ter clientes especiais, ou enviar promoções em seus aniversários, cabendo ao estabelecimento a decisão.



**Sismannu**

Telefone:

**Confirme ou altere os dados abaixo**

Nome:

Data Nascimento (ex. 10/01/2006):

Enviar

Figura 30 – Tela de Cadastro

Após enviar todos os dados, a tela principal ou Tela de Cardápio (**Figura 31**) será aberta. Nesta tela o cliente verá os produtos que o estabelecimento têm a oferecer e escolher a quantidade de cada. Para fins de designer, as comidas estão no lado esquerdo e as bebidas no direito. Outro objetivo do projeto é proporcionar ao cliente a liberdade de escolha na hora que quiser sem precisar sofrer uma pressão por parte do Garçom como acontece em alguns lugares. Uma vez logado no sistema, ele tem o tempo que quiser para escolher.

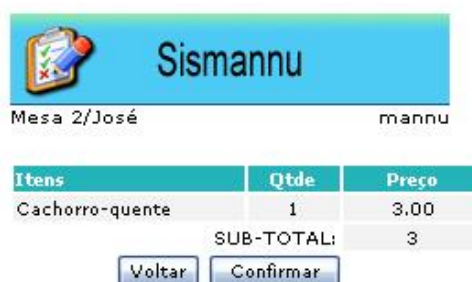


Item	Qtde
Cachorro-quente	0
Filé ao molho madeira	0
Prato da casa	0
Talharim	0
Coca-Cola	0
Guaraná	0
Suco de Laranja	0
Suco de Uva	0

Conta Parcial Confirmar

Figura 31 – Tela de Cardápio

Existem também na Tela de Cardápio, as informações: número da mesa, nome do garçom responsável e o nome do cliente. Ao escolher o item e a quantidade, o cliente selecionará o *Confirmar* e em seguida aparecerá uma tela contendo os valores dos produtos, uma vez que não ficaria graficamente bom na tela principal.

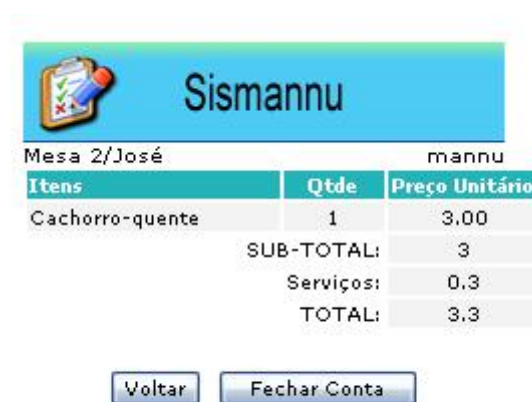


Itens	Qtde	Preço
Cachorro-quente	1	3,00
SUB-TOTAL:		3

Voltar Confirmar

Figura 32 – Tela de Valores

Depois de verificado o valor, o cliente tem a opção de voltar para a tela principal e escolher outro item, ou efetivar o pedido acionando o botão *Confirmar* novamente. Ao confirmar, o pedido é automaticamente enviado à tabela de Pedidos e atualizado nos outros dois módulos. Assim, a tela retornará novamente para o Cardápio (**Figura 31**). E a qualquer momento ele tem a opção de fazer um novo pedido ou ver a conta parcial.



**Sismannu**

Mesa 2/José      mannu

Itens	Qtde	Preço Unitário
Cachorro-quente	1	3,00
SUB-TOTAL:		3
Serviços:		0,3
TOTAL:		3,3

Figura 33 – Conta Parcial

Na tela Conta Parcial (**Figura 33**), o cliente poderá retornar ao Cardápio ou solicitar o encerramento da conta, representado pela atividade *Fechar Conta*. Ao solicitar, o Caixa recebe a informação pela tabela Conta e fecha a conta. Neste momento, o sistema fica bloqueado (**Figura 34**) para essa mesa, impedindo que o cliente faça um pedido que acabará não entrando na conta final.





Figura 34 – Conta Fechada

O sistema fica bloqueado para o cliente, mas habilitado para o garçom. Pode ter ocorrido de o cliente ter solicitado o encerramento erroneamente da conta, então o garçom, com sua senha de acesso, pode re-ativar a conta voltando ao seu último estado. Caso o cliente tenha pagado a conta, o Caixa fará a Liberação da mesa e o garçom abrirá uma nova conta quando um cliente chegar.

O fluxo de funcionamento para este módulo pode ser entendido também pela **Figura 35**.

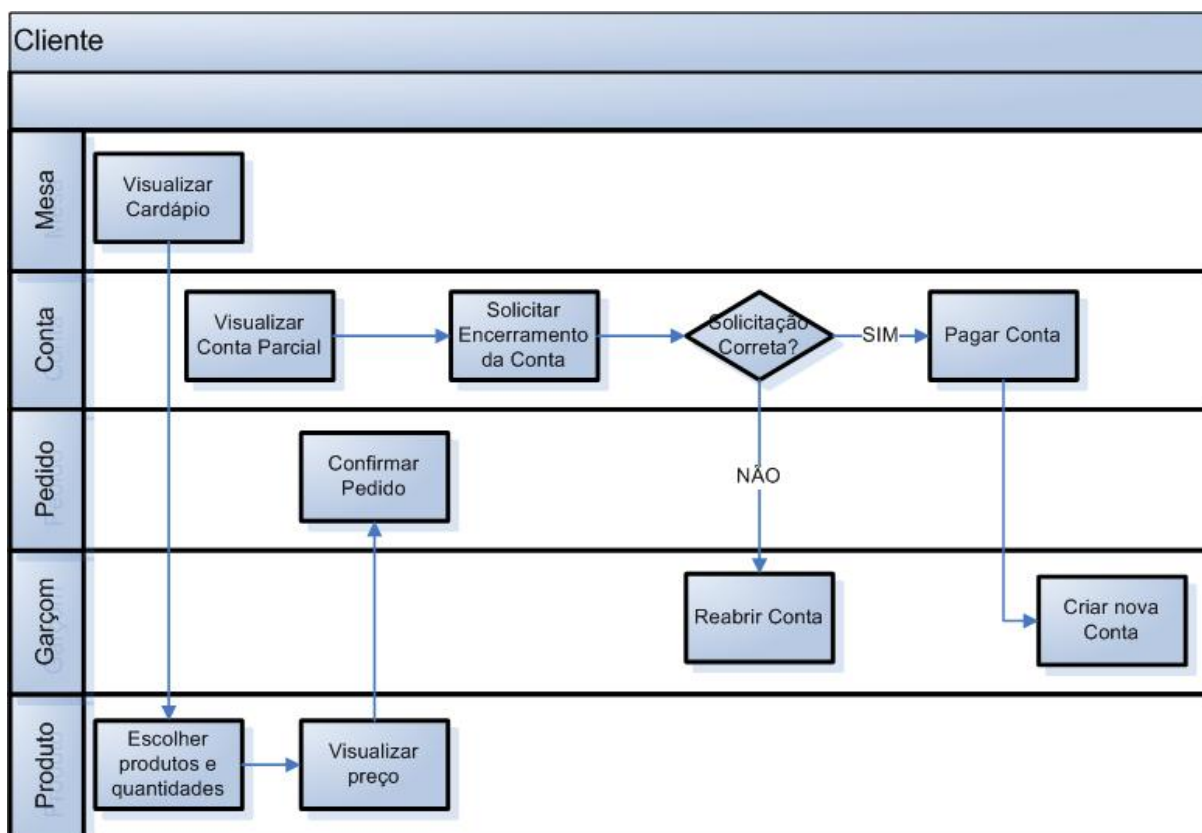


Figura 35 – Fluxograma do Cliente

Como o projeto foi desenvolvido para fins acadêmicos, a segurança de informação não foi contemplada.

### 3.4 INTEGRAÇÃO

Para a integração entre todos os dispositivos e módulos do sistema, foi adotada uma separação entre:

- Caixa / Cliente;
- Caixa / Balcão;
- Balcão / Cozinha.

#### 3.4.1 Caixa / Cliente

A integração Caixa / Cliente é feita entre o servidor central, que é o Módulo de Administração – Caixa, e o PDA que estará localizado em cada mesa do estabelecimento.

No desenvolvimento do projeto foram testadas duas conexões para este grupo: cabo USB e conexão Wireless. Sendo as duas alcançadas com sucesso, porém com algumas restrições.

Na conexão pelo cabo USB, foi utilizado o cabo de sincronia entre o PC e o PDA que vem na compra do dispositivo, sendo que este cabo não permite somente a transferência de dados e voz, permite também a conexão com a intranet que foi estabelecida na implementação do sistema. Ou seja, pelo cabo USB o sistema pode ser acessado corretamente. Na compra do PDA foi escolhido um Palm que tivesse o sistema operacional Windows, pois teria uma menor chance de erro na comunicação, devido ao servidor possuir o mesmo sistema operacional, e também porque o módulo Cliente foi desenvolvido para o navegador Internet Explorer, sendo um navegador comum ao servidor e ao PDA. Entretanto, não é possível afirmar que não executaria em outras marcas ou sistemas operacionais, foi apenas uma escolha para a implementação acadêmica.

Na conexão Wireless foram testadas duas maneiras: conexão com a placa de rede wireless do PC e o wireless do PDA, e os dois conectados a um roteador wireless. Na primeira tentativa, a idéia era configurar o PC para ser o ponto-de-acesso, porém ao terminar a configuração o sistema ficava apenas aproximadamente cinco segundos conectado, tendo que ser restabelecida a conexão. Após testes de configuração não obtidos com êxitos, visto que não estava no escopo primordial do projeto o estudo em conexão wireless, foi testada uma conexão entre os dois pontos com um terceiro dispositivo fazendo o papel de ponto-de-acesso, o roteador wireless, no qual os outros dois se conectam sem quedas na rede. Nessa segunda tentativa o resultado foi obtido com sucesso, sendo assim, para fins de implementação acadêmica, será utilizado um roteador wireless que é o responsável em fazer toda a comunicação entre o módulo Caixa e o Cliente.

### **3.4.2 Caixa / Balcão**

Para o grupo Caixa / Balcão, foi escolhida a conexão por cabo de rede, devido à facilidade na implementação estrutural, menor custo comparado à placa de rede wireless e aumentar a variedade de tecnologias, mostrando que existem outras formas para conexão, obtendo uma maior mobilidade, sendo necessária então uma configuração na rede doméstica do estabelecimento.

É possível também implementar a integração com uma conexão wireless, ficando a critério do estabelecimento, uma vez que o ponto-de-acesso já estará disponível para o grupo Caixa / Cliente, faltando apenas configurações de IP, máscara de sub-rede e gateway padrão, conforme utilizado para os demais componentes da rede.

### 3.4.3 Balcão / Cozinha

A integração entre o Balcão e a Cozinha é dividida em três partes:

- Java;
- Smartie;
- Porta Paralela.

O programa em Java (ver apêndice III), conforme citado no item **3.2.2.1** Erro! Fonte de referência não encontrada., será executado assim que o PC for ligado. A rotina funciona independente do sistema como um todo, ou seja, o Java não precisa do servidor web ativado para desenvolver suas tarefas, por esse motivo foi escolhida essa linguagem de programação, pois qualquer erro que possa acontecer no sistema não afetará a preparação do produto pela cozinha, bastando apenas que o computador esteja ligado e o banco de dados esteja ativo.

A rotina desenvolvida ordenará os resultados obtidos pela coluna Ped\_Data\_Reg na Tabela Pedido com data mais antiga, ou seja, o primeiro pedido que foi feito será o primeiro pedido da fila, sendo que para a recuperação dos dados existem alguns pré-requisitos como por exemplo: a conta e o pedido estarem com status aberto, e o produto ser do tipo comida que foi cadastrado na tabela representado pelo número zero. O Java abre uma conexão com o MySQL e passa a seguinte *query*:

```
SELECT prod.Prod_nome, php.Has_Qtde as QTDE
FROM Conta cta, Pedido ped, Pedido_has_Produto php, Produto prod
WHERE cta.Cta_Id = ped.Cta_Id
AND ped.Ped_ID = php.Ped_ID
AND php.Prod_ID = prod.Prod_ID
AND prod.Prod_Tipo = 0
AND cta.Cta_Status = 'A'
AND ped.Ped_Estado = 'A'
ORDER BY ped.Ped_Data_Reg asc, prod.Prod_nome asc;
```

Ao retornar os dados, o Java gerará um arquivo texto contendo os produtos e quantidades correspondentes a cada pedido e os colocará em linhas seqüenciais ordenados ascendentemente pela data mais antiga e em seguida pelo nome dentro do mesmo pedido, conforme ilustrado abaixo (**Figura 36**). O pedido é retirado da fila quando o responsável pelo balcão realizar a tarefa de *Liberar Pedido*, pois assim o pedido estará com o status fechado, e a query só busca pedidos abertos.

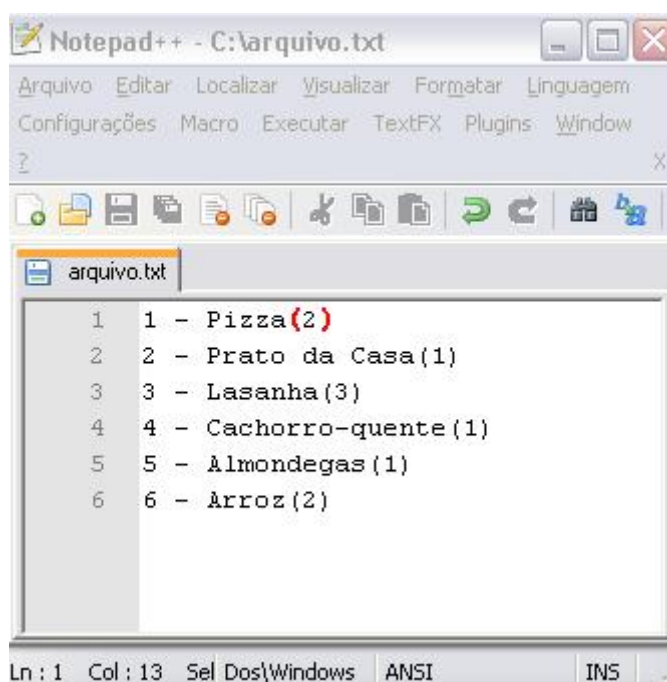


Figura 36 – Arquivo Texto

O arquivo será gravado no diretório **C:** do PC do balcão e a cada cinco segundos a rotina será atualizada, pois quando a coluna *Ped\_Data\_Fim* for inserida no banco pela ação *Liberar Pedido* do módulo do Balcão, uma nova ordem será respeitada, criando assim o ciclo de atualizações dos pedidos e arquivos.

O freeware Smartie, é o responsável por buscar o arquivo texto criado pelo Java e mostrá-lo no display. Ou seja, essa ferramenta permite a comunicação efetiva entre a ordem de pedidos e a ilustração deles pelo display. O software foi adotado para diminuir os gastos, pois é um freeware e por possibilitar uma boa implementação entre um arquivo texto e o display LCD.

Na tela de configuração do software (**Figura 37**), foi adotada a opção que permite a rolagem dos itens e definido que seriam apresentados os três primeiros na primeira linha do display e os três últimos na segunda linha. Como o display adotado

possui duas linhas e dezesseis colunas, foram escolhidos somente seis produtos por vez para ter uma melhor visualização sem poluição visual no display, evitando muita informação ao cozinheiro, e permitindo uma maior habilidade caso dentre esses seis pedidos alguns sejam parecidos, podendo ser adiantados em uma única vez.

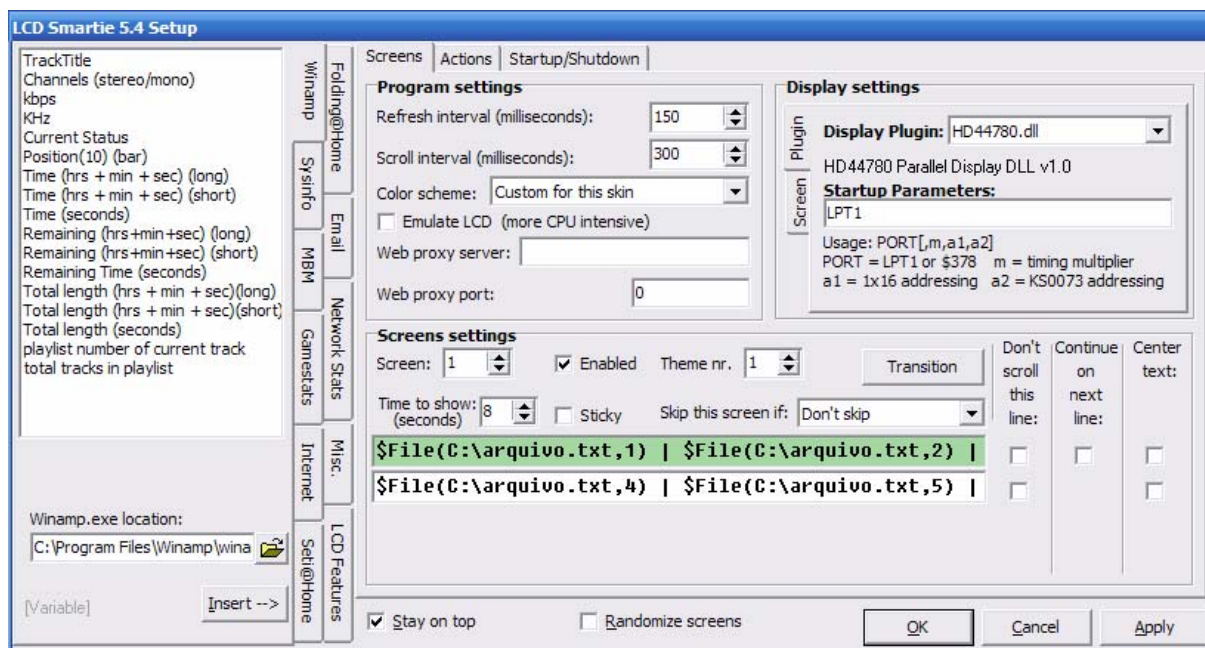


Figura 37 – Tela de Configuração do Smartie

O Smartie possui uma tela que simula o funcionamento configurado em um LCD. Esta tela foi usada no período de testes independentes, ou seja, enquanto o desenvolvimento estava sendo isolado em cada parte. A disponibilidade do simulador ajudou enquanto o display LCD não estava pronto, permitindo então que a rotina em Java pudesse ser testada sem a necessidade de visualização somente no display.

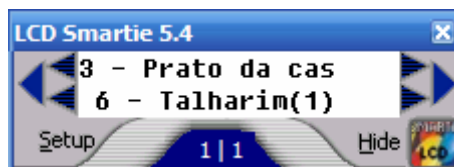


Figura 38 – Simulador do Smartie

A comunicação final entre o PC e o display foi feita pela porta paralela, e o responsável também pela comunicação é o Smartie. O software possui o algoritmo

internamente que provê a o reconhecimento do PC ao display pela porta paralela e recebe também as informações de cada pino.

O display LCD utilizado no projeto foi comprado no Rio Grande do Sul, com uma loja que possui convênio com a Universidade Federal do Estado [LASERSHAPE]. Em um primeiro momento, o LCD foi conectado à porta paralela pela sondagem nos pinos, porém ao testar ligando ao computador, a voltagem não estava apropriada e o display queimou. Então, para implementação em banca, foi adquirido um kit completo com o display LCD e a porta paralela, sendo que a montagem do mesmo não era o foco do projeto.

A série do display utilizado é a HD44780 16x2 e o Smartie é o encarregado de toda comunicação entre porta paralela, display, PC e arquivo texto, sendo que ele é compatível com outras séries do LCD, portanto, o software pode ser empregado em outros projetos que possuam outro tipo de display LCD e outra área de atuação.

O fluxo de funcionamento para esta interação pode ser entendido também pela **Figura 39**.

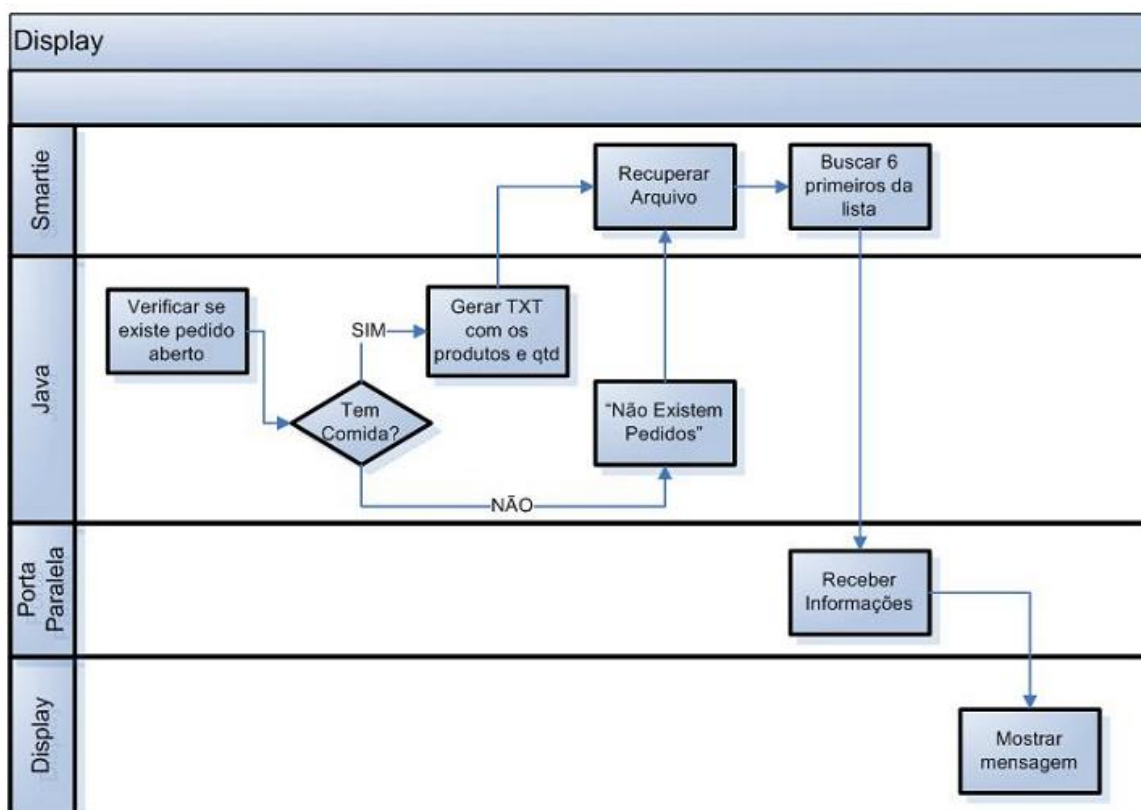


Figura 39 – Fluxograma de Interação



Após todas as fases concluídas na execução do projeto, a implementação final, que será apresentada em banca, possui os seguintes componentes: o PDA, o display, um notebook atuando como o servidor/Caixa e um desktop atuando como o terminal do Balcão (**Figura 40**).

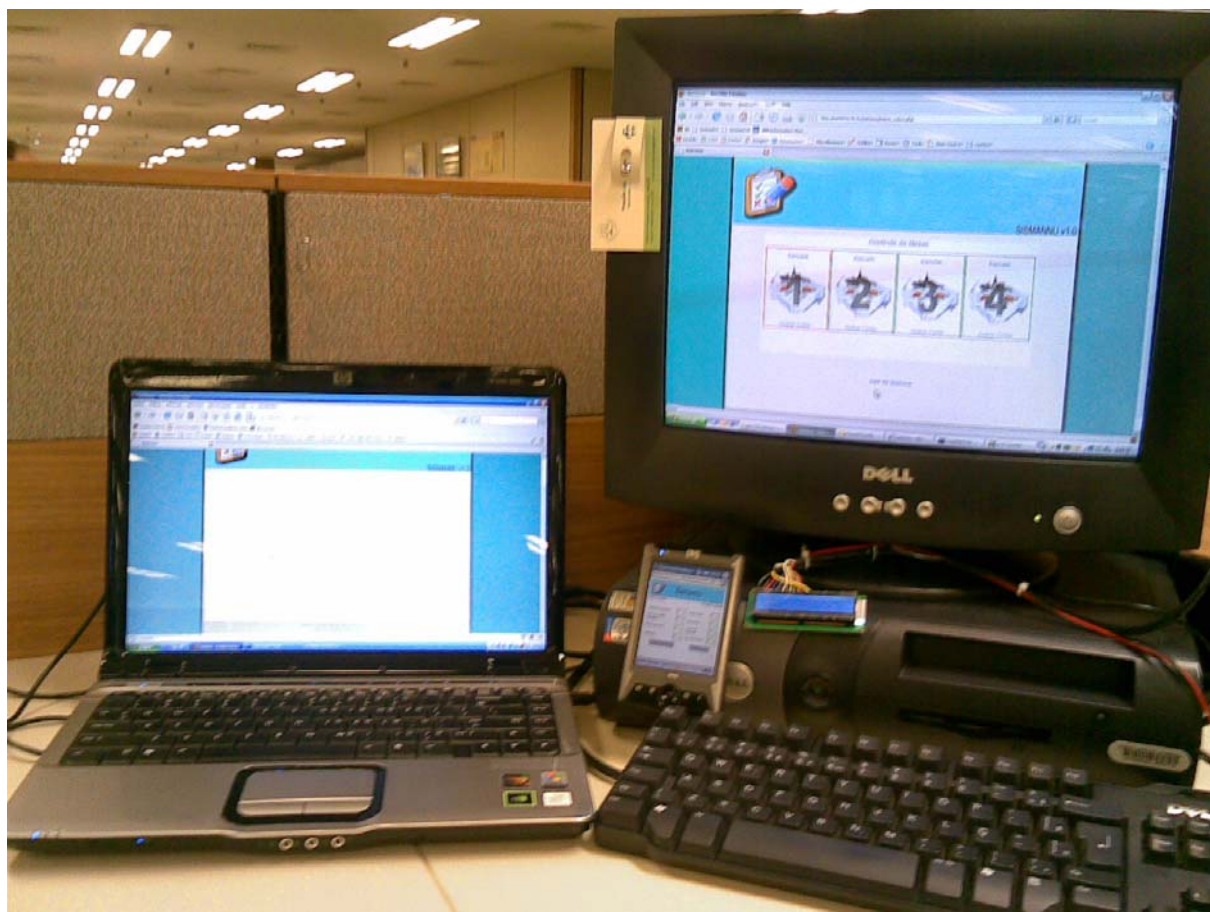


Figura 40 – Implementação Final



## CAPÍTULO 4 – CONCLUSÃO

O SisMannu é um sistema projetado para permitir o gerenciamento de pedidos realizados por clientes em um restaurante. A partir de um PDA, localizado na mesa do cliente, o mesmo poderá solicitar seus pedidos e acompanhar o valor de sua conta. Aliado a isso o responsável pelo estabelecimento saberá todas as ocorrências de pedidos e escolha de produtos através de qualquer computador em sua rede local, permitindo uma integridade entre os demais funcionários.

Um dos objetivos é proporcionar à sociedade um meio de interação com tecnologias atuais, pois nem sempre todos têm o acesso à esta realidade informatizada, e este foi um dos assuntos muito discutidos no decorrer do curso de Engenharia da Computação, a inclusão digital na sociedade.

A proposta do projeto possui um foco interessante e eficiente para as empresas que querem agilizar e investir na comodidade do cliente. Com as ferramentas escolhidas não será muito caro o investimento, somente na parte do hardware, por isso o software foi escolhido com o menor custo possível, porém com uma configuração adequada para uma boa performance.

A realização do projeto foi alcançada com êxito em todas as suas partes: modelagem, sistema web, hardware e integração. Atualmente não encontramos no mercado projetos com tantas soluções integradas, portanto foi alcançado o objetivo principal que é a sinergia de tecnologias.

### 4.1 FUTUROS PROJETOS

O projeto pode ser utilizado tanto por alunos da Engenharia da Computação quanto da Ciência da Computação. São sugeridas as seguintes melhorias:

- Desenvolvimento de um módulo para solicitação de pedidos pela internet ou acesso wap;
- Criação de um circuito com microprocessador/microcontrolador localizado na cozinha, onde serão armazenados os pedidos e o responsável terá a possibilidade de percorrer entre eles e encerrar quando o mesmo estiver pronto;

- Implementação medidas de segurança de acesso que garantam a proteção de informações do estabelecimento;
- Aprimoramento no módulo do Caixa com funções de gerar gráficos com o balanço diário, semanal ou mensal;
- Implementação de um módulo que faça o controle de estoque verificando os produtos disponíveis e enviando avisos ao gerente quando estiver acabando algum item.

## Referências Bibliográficas

- MASLAKOWSKI, Mark, *Teach Yourself MySQL in 21 Days*, Indianapolis - EUA, 2000
- NIEDERAUER, Juliano, *Guia de Consulta Rápida PHP5 – Inclui as versões 4 e 5*, 2. ed. São Paulo: Novatec Editora, 2005.
- NIEDERAUER, Juliano, *Guia de Consulta Rápida XHTML*, São Paulo: Novatec Editora, 2002.
- OLIVEIRA, Carlos Antônio José, *Faça um Site - PHP4 com Base de Dados MySQL Orientado por Projeto*, São Paulo: Érica, 2001.
- SANTOS, Antonio Raimundo dos, *Metodologia Científica: a construção do conhecimento*. 3. ed. Rio de Janeiro: DP&A editora, 2000.
- TORRES, Gabriel Hardware Curso Completo. 4. ed. Rio de Janeiro: Axcel, 2001.

### SITES

ACHOUR, Mehdi - **“PHP Manual”**

<http://www.php.net/manual/en/index.php>

Acesso em 17/04/2006

APACHE FOUNDATION – **“Apache HTTP Project”**

<http://httpd.apache.org/>

Acesso em 23/04/2006

ADAPTIVE PATH – **“AJAX: A New Approach to Web Applications”**

<http://adaptivepath.com/publications/essays/archives/000385.php>

Acesso em 17/04/2006

GUIA DO HARDWARE – **“Ponto-de-Acesso”**

<http://www.guiadohardware.net/termos/ponto-de-acesso>

Acesso em 10/04/2006

**GUJ – “Aprendendo a Manipular Arquivos”**

<http://www.guj.com.br/java/tutorial.artigo.13.1.guj>

Acesso em 15/10/2006

**IMASTERS – “JavaScript”**

<http://www.imasters.com.br/secao/javascript>

Acesso em 20/04/2006

**INFOWESTER – “Apache HTTP Server Project”**

<http://www.infowester.com/servapach.php>

Acesso em 23/04/2006

**LASERSHAPE – “Display LCD + Porta Paralela”**

<http://www.lasershape.com.br>

Acesso em 29/01/2007

**LCD – “HD44780 – Based LCD Modules”**

<http://www.doc.ic.ac.uk/~ih/doc/lcd/>

Acesso em 10/02/2006

**LCD – “LCD Smartie”**

<http://lcdsmartie.sourceforge.net/>

Acesso em 10/02/2006

**NETCRAFT – “April 2006 Web Server Survey”**

[http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)

Acesso em 25/04/2006

**PHPBRASIL – “PHP”**

<http://www.phpbrasil.com/>

Acesso em 03/03/2006

**PHP – “PHP: Hypertext Preprocessor”**

<http://www.php.net/>

Acesso em 03/03/2006

**PORTAL JAVA – “Artigos e Tutoriais”**

<http://www.portaljava.com/home/modules.php?name=Content>

Acesso em 10/10/2006

**PROTOTYPE – “LEARN”**

<http://www.prototypejs.org/learn>

Acesso em 05/05/2006

**TABLELESS – “Aprenda”**

<http://tableless.com.br/aprenda/>

Acesso em 27/05/2006

**W3C – “HTML 4.01 Specification”**

<http://www.w3.org/TR/REC-html40/cover.html>

Acesso em 17/04/2006

**WIKIPEDIA – “LCD”**

<http://pt.wikipedia.org/wiki/LCD>

Acesso em 02/02/2007

**WIKIPEDIA – “COMPUTADOR PESSOAL”**

[http://pt.wikipedia.org/wiki/Computador\\_pessoal](http://pt.wikipedia.org/wiki/Computador_pessoal)

Acesso em 10/04/2006

**WIKIPEDIA – “COMPUTADOR PESSOAL”**

<http://pt.wikipedia.org/wiki/PDA>

Acesso em 10/04/2006

WIKIPEDIA – “**COMPUTADOR PESSOAL**”

<http://pt.wikipedia.org/wiki/Automa%C3%A7%C3%A3o>

Acesso em 10/04/2006

WIKIPEDIA – “**CSS**”

[http://pt.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://pt.wikipedia.org/wiki/Cascading_Style_Sheets)

Acesso em 17/04/2006

WIKIPEDIA – “**MySQL**”

<http://pt.wikipedia.org/wiki/MySQL>

Acesso em 20/04/2006

# APÊNDICE I

## INDEX.PHP

```
<?
//$ambiente = $_SERVER["HTTP_USER_AGENT"];
//if($ambiente)

$ambiente = strtoupper($_SERVER["HTTP_USER_AGENT"]);
$ocorrencia = strpos($ambiente, 'PPC');
if($ocorrencia != ""){
    header("Location: pocket_index.php");
}else{
    header("Location: index_pc.php");
}
?>
```

## INDEX\_PC.PHP

```
<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include($root."/template/header.php");?>
<script type="text/javascript">
function erro(){
    $("erro_login").style.visibility= "visible";
    $("erro_login").innerHTML = url_decode("<?=rawurlencode($msgErro[0])?>");
    Effect.BlindDown('erro_login');
    var desaparecer = function(){
        return new Effect.Fade('erro_login');
    };
    timeoutStatusId = window.setTimeout(desaparecer, 5000);
}
function enter(e){
    if(e.keyCode == 13){
        logar();
    }
}
</script>
<div id="login">
<div class="cabecalho">
```

```

        <span>LOGIN</span>
    </div>
    <div id="login_corpo">
        <form id="form_login" action="controlador.php?action=autenticar" method="post">
            <div>
                <div class="label_login"><span>Login:</span></div>
                <div class="input_login"><input name="txtLogin" type="text"
onkeypress="enter(event);"/></div>
            </div>
            <div>
                <div class="label_login"><span>Senha:</span></div>
                <div class="input_login"><input name="txtSenha" type="password"
onkeypress="enter(event);"/></div>
            </div>
            <div class="botao_login">
                <input type="button" name="name" value="Entrar" onclick="logar();"
onkeypress="enter(event);"/>
            </div>
        </form>
    </div>
    <div id="resposta_lendo" style="visibility:hidden;text-align:center">
        
    </div>
    <div id="erro_login" style="visibility:hidden;display:none;"><?=$msgErro[0]?></div>
</div>
<? if(isset($_GET["msg"])) {
    if($_GET["msg"] == "erro"){?>
        <script language="javascript">acessoNegado("<?=$msgErro[1]?>");</script>
    <? }else if($_GET["msg"] == "logout"){
        logout();
    }
} ?>
<?include($root."/template/footer.php");?>

```

## POCKET\_INDEX.PHP

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include($root."/template/header_pocket.php");?>
<div id="loginPocket">
    <form action="controlador.php?action=escolherMesa" method="POST">

```



```

        <p>Escolha uma mesa</p>
        <p><?
require($_SERVER["DOCUMENT_ROOT"]."/sistema/modulos/mesa/mesas.php"); ?></p>
        <p><input type="submit" value="Confirmar"/></p>
        <p></p>
    </form>
</div>
<?include($root."/template/footer_pocket.php");?>

```

## CONFIG\_SERVER.PHP

```

<?
$documentRoot = "http://".$_SERVER["HTTP_HOST"] . "/sistema";
$root = $_SERVER["DOCUMENT_ROOT"]."/sistema";
$senhaGarcom = "mannu";

$msgErro[0] = "<span>Usuário/Senha inválidas</span>";
$msgErro[1] = "<span>Acesso negado</span>";

function logout(){
    session_start("siscozinha");
    session_unregister("login");
    session_unregister("perfil");
    session_destroy();
}
?>

```

## CONTROLADOR.PHP

```

<?php
/*
 * Criado em 26/07/2006
 * Autor(a): Mannuela Cruz
 * Página que gerencia as chamadas para os módulos
 */
$modulo = $_GET["action"];
$caminhoModulo = $_SERVER["DOCUMENT_ROOT"]."/sistema/modulos";
switch($modulo){
    case "autenticar":

```

```

include $caminhoModulo."/seguranca/autenticacao.php";
break;
case "comboGarcom":
include $caminhoModulo."/mesa/comboGarcom.php";
break;
case "alterarGarcom":
include $caminhoModulo."/mesa/alterarGarcom.php";
break;
case "getGarcomMesa":
include $caminhoModulo."/mesa/garcomMesa.php";
break;
case "detalheMesa":
include $caminhoModulo."/mesa/detalheMesa.php";
break;
case "statusMesas":
include $caminhoModulo."/mesa/statusMesas.php";
break;
case "getPedidos":
include $caminhoModulo."/pedido/listaPedidos.php";
break;
case "fecharConta":
include $caminhoModulo."/conta/fecharConta.php";
break;
case "liberarMesa":
include $caminhoModulo."/conta/liberarContaMesa.php";
break;
case "getMesas":
include $caminhoModulo."/mesa/mesas.php";
break;
case "escolherMesa":
include $caminhoModulo."/mesa/escolherMesa.php";
break;
case "getClient":
include $caminhoModulo."/cliente/pocket_getCliente.php";
break;
case "realizarPedido":
include $caminhoModulo."/pedido/pocket_realizarPedido.php";
break;
case "pocketLiberarMesa":

```

```

        include $caminhoModulo."/mesa/pocket_liberar_mesa.php";
        break;
        case "statusPedido":
            include $caminhoModulo."/pedido/statusPedido.php";
            break;
        case "detalheMesaBalcao":
            include $caminhoModulo."/mesa/detalheMesaBalcao.php";
            break;
        case "liberarPedido":
            include $caminhoModulo."/pedido/liberarPedidoMesa.php";
            break;
    }
?>

```

## **MENU\_BALCAO.PHP**

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php")?>
<?include("seguranca.php"); ?>
<?include($root."/template/header.php");?>

<script type="text/javascript"><?//?>
var verificarStatusTimeout;
function exibirMesa(idMesa){
    var loading_mesa = $("loading_mesa");
    Element.setStyle(loading_mesa,{visibility: "visible"});
    Element.setStyle(loading_mesa,{display: "inline"});

    var request = new AjaxRequest("<?=$documentRoot?>/controlador.php");
    request.addParameter("action", "detalheMesaBalcao");
    request.addParameter("mesaID", idMesa);
    request.setTextResponseHandlerFunction(mesaCarregada);
    request.send();
}

function mesaCarregada(resposta){
    var loading_mesa = $("loading_mesa");
    var infoMesa = $("info_mesa");

    Element.setStyle(loading_mesa,{visibility: "hidden"});
    Element.setStyle(loading_mesa,{display: "none"});

```

```

        infoMesa.innerHTML = url_decode(resposta);
        Element.setOpacity(infoMesa, 0.90);
        Element.setStyle(infoMesa,{visibility: "visible"});
        Effect.BlindDown('info_mesa');
    }
    function fecharDetalheMesa(){
        Effect.BlindUp('info_mesa');
        $("info_mesa").innerHTML = "";
    }
    function verificarStatus(){
        if($("info_mesa").style.visibility != "visible" || $("info_mesa").innerHTML == ""){
            var request = new AjaxRequest("<?=$documentRoot?>/controlador.php");
            request.addParameter("action", "statusPedido");
            request.setTextResponseHandlerFunction(statusMesas);
            request.send();
        }
        verificarStatusTimeout = window.setTimeout(verificarStatus,5000);
    }
    function statusMesas(resposta){
        var mesas = resposta.split(";");

        for(i=0;i<mesas.length;i++){
            if(mesas[i] != ""){
                mesa = mesas[i].split("=");
                $(mesa[0]).className = mesa[1];
                var textoObs = "";
                if(mesa[2] != 0){
                    textoObs = "Ordem: <b>" + mesa[2] + "</b>";
                }
                var obs = $("obs_" + mesa[0]);
                obs.innerHTML = textoObs;
                if(mesa[1] == "mesaPause"){
                    var imagem = $(mesa[0]);
                    Effect.Pulsate(imagem,{duration:2});
                }
            }
        }
    }
}

function liberarPedido(){

```

```

        var idPedido = $F("idPedidoAtual");
        var request = new AjaxRequest("<?=$documentRoot?>/controlador.php");
        request.addParameter("action", "liberarPedido");
        request.setTextResponseHandlerFunction(resultadoLiberarPedido);
        request.addParameter("pedidoID", idPedido);
        request.send();
    }
    function resultadoLiberarPedido(resposta){
        alert("Pedido Encerrado");
        fecharDetalheMesa();
    }
<??>
</script>

        <div id="loading_mesa" style="visibility:hidden;display:none;"></div>
        <div id="info_mesa" style="visibility:hidden;display:none;"></div>
        <div id="conteudo_caixa">
            <div id="caixa_titulo"><span>Controle de Mesas</span></div>
            <div id="balcao">
                <div id="mesa1" class="mesaOn">
                    <p><br/>
                    <span id="obs_mesa1"></span>
                    </p>
                </div>
                <div id="mesa2" class="mesaOn">
                    <p><br/>
                    <span id="obs_mesa2"></span>
                    </p>
                </div>
                <div id="mesa3" class="mesaOn">
                    <p><br/>
                    <span id="obs_mesa3"></span>
                    </p>
                </div>
                <div id="mesa4" class="mesaOn">

```

```

                                <p><br/>
                                <span id="obs_mesa4"></span>
                                </p>
                                </div>
                            </div>
                        </div>
                    <script type="text/javascript">
                        verificarStatus();
                    </script>
                <?include($root."/template/footer.php");?>

```

## **MENU\_CAIXA.PHP**

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include("seguranca.php"); ?>
<?include($root."/template/header.php");?>

<script type="text/javascript"><?//?>
function mostrarBalao(refPosEI, idMesa) {
    var balaoDiv = $("balao");
    Element.setOpacity(balaoDiv, 0.90);
    Element.setStyle(balaoDiv, {visibility: "hidden"});
    var posicaoElemento = getPosition(refPosEI);
    Element.setStyle(balaoDiv, {top: (posicaoElemento.y - 150) + "px", left:
(posicaoElemento.x - 120) + "px", visibility: "visible"});

    /* Chamando a combo */
    $("img_lendo").style.visibility= "visible";
    Element.setStyle($("balao_resposta"), {visibility: "hidden"});
    var request = new AjaxRequest("<?=$documentRoot?>/controlador.php");
    request.addParameter("action", "comboGarcom");
    request.addParameter("mesaID", idMesa);
    request.setTextResponseHandlerFunction(lerComboGarcom);
    request.send();
}
function lerComboGarcom(resposta){
    $("img_lendo").style.visibility= "hidden";
    $("balao_conteudo").innerHTML = url_decode(resposta);
}

```

```

function alterarGarcom(){
    var form = $("form_garcom");
    $("img_lendo_combo").style.visibility= "visible";
    ajaxReq = new AjaxRequest(form.action);
    ajaxReq.addForm(form);
    ajaxReq.setTextResponseHandlerFunction(resultadoAlterarGarcom);
    ajaxReq.send();
}

var timeoutResultadoGarcom;
function resultadoAlterarGarcom(resposta){
    $("img_lendo_combo").style.visibility= "hidden";
    $("balao_resposta").style.visibility= "visible";
    Effect.Appear('balao_resposta');
    var desaparecer = function(){
        return new Effect.Fade('balao_resposta');
    };
    timeoutResultadoGarcom = window.setTimeout(desaparecer, 3000);
}

function exibirMesa(idMesa){
    esconderBalao();

    var loading_mesa = $("loading_mesa");
    Element.setStyle(loading_mesa,{visibility: "visible"});
    Element.setStyle(loading_mesa,{display: "inline"});

    var request = new AjaxRequest("<?=$documentRoot?>/controlador.php");
    request.addParameter("action", "detalheMesa");
    request.addParameter("mesaID", idMesa);
    request.setTextResponseHandlerFunction(mesaCarregada);
    request.send();
}

function mesaCarregada(resposta){
    var loading_mesa = $("loading_mesa");
    var infoMesa = $("info_mesa");

    Element.setStyle(loading_mesa,{visibility: "hidden"});
    Element.setStyle(loading_mesa,{display: "none"});

```

```

        infoMesa.innerHTML = url_decode(resposta);
        Element.setOpacity(infoMesa, 0.90);
        Element.setStyle(infoMesa,{visibility: "visible"});
        Effect.BlindDown('info_mesa');
    }
    var verificarStatusTimeout;
    function verificarStatus(){
        if($("#info_mesa").style.visibility != "visible" || $("#info_mesa").innerHTML == ""){
            var request = new AjaxRequest("<?=$documentRoot?>/controlador.php");
            request.addParameter("action", "statusMesas");
            request.setTextResponseHandlerFunction(statusMesas);
            request.send();

        }
        verificarStatusTimeout = window.setTimeout(verificarStatus,5000);
    }
    function statusMesas(resposta){
        var mesas = resposta.split(";");
        for(i=0;i<mesas.length;i++){
            $("#mesa"+(i+1)).className = mesas[i];
            if(mesas[i] == "mesaPause"){
                var imagem = $("#mesa" + (i+1));
                Effect.Pulsate(imagem,{duration:2});
            }
        }
    }
}
<?//?>
var timeoutCarregarConta;
function carregarConta(idConta){
    Effect.Fade('info_mesa_conteudo');
    var aparecerConta = function(){
        var loading_mesa = $("#loading_mesa");
        Element.setStyle(loading_mesa,{visibility: "visible"});
        Element.setStyle(loading_mesa,{display: "inline"});

        var request = new AjaxRequest("<?=$documentRoot?>/controlador.php");
        request.addParameter("action", "getPedidos");
        request.setTextResponseHandlerFunction(apresentarConta);
        request.addParameter("contaID", idConta);
    }
}

```



```

        request.send();
    };
    timeoutCarregarConta = window.setTimeout(aparecerConta, 1000);
}
var contaAnterior;

function apresentarConta(dadosConta){
    if(correcao(dadosConta)){
        var loading_mesa = $("loading_mesa");
        Element.setStyle(loading_mesa,{visibility: "hidden"});
        Element.setStyle(loading_mesa,{display: "none"});
        contaAnterior = $("info_mesa_conteudo").innerHTML;
        $("info_mesa_conteudo").innerHTML = url_decode(dadosConta);
        Effect.Appear('info_mesa_conteudo');
    }
}

var timeoutVoltarConta;
function voltarConta(){
    if(correcao(contaAnterior)){
        Effect.Fade('info_mesa_conteudo');
        var aparecerAnterior = function(){
            $("info_mesa_conteudo").innerHTML = contaAnterior;

            Effect.Appear('info_mesa_conteudo');
        };
        timeoutVoltarConta = window.setTimeout(aparecerAnterior, 1000);
    }
}

function fecharConta(){
    var confirma = $("confirmar_fechamento");
    Element.setStyle(confirma,{visibility: "visible"});
}

function confirmarConta(opcao){
    if(opcao == "1"){
        var idConta = $("idContaAtual").value;
        var request = new AjaxRequest("<?=$documentRoot?>/controlador.php");
        request.addParameter("action", "fecharConta");
        request.setTextResponseHandlerFunction(resultadoFecharConta);
    }
}

```

```

        request.addParameter("contaID", idConta);
        request.send();
        //alert(idConta);
    } else {
        var confirma = $("confirmar_fechamento");
        Element.setStyle(confirma,{visibility: "hidden"});
    }
}

function resultadoFecharConta(resposta){
    alert("Conta fechada com sucesso");
    var confirma = $("confirmar_fechamento");
    var btnFecharConta = $("fecharConta");
    Element.setStyle(confirma,{visibility: "hidden"});
    btnFecharConta.disabled = true;
    btnFecharConta.className = "botaoOff";
}

function liberarMesa(){
    var btnFecharConta = $("fecharConta");
    if(btnFecharConta.disabled){
        var idConta = $("idContaAtual").value;
        var request = new AjaxRequest("<?=$documentRoot?>/controlador.php");
        request.addParameter("action", "liberarMesa");
        request.setTextResponseHandlerFunction(resultadoLiberarMesa);
        request.addParameter("contaID", idConta);
        request.send();
    } else {
        alert("Feche a conta primeiro");
    }
}

function resultadoLiberarMesa(resposta){
    alert("Mesa Liberada");
    fecharDetalheMesa();
}

function fecharDetalheMesa(){
    Effect.BlindUp('info_mesa');
    $("info_mesa").innerHTML = "";
}

<??>
</script>

```

```

<div id="confirmar_fechamento" >
    <p><span>Deseja realmente fechar a conta deste
cliente?</span></p>
    <p>
        <span><input type="button" name="btnSim" value="Sim"
onclick="confirmarConta(1);"/></span>
        <span><input type="button" name="btnNao" value="Não"
onclick="confirmarConta(2);"/></span>
    </p>
</div>
<div id="loading_mesa" style="visibility:hidden;display:none;"></div>
<div id="info_mesa" style="visibility:hidden;display:none;"></div>
<div id="balao">
    <div id="balao_topo"></div>
    <div id="balao_fechar" onclick="esconderBalao();"></div>
    <div id="balao_div"></div>
    <div id="balao_conteudo"></div>
    <div id="balao_fim">
        <center><span id="balao_resposta"
style="visibility:hidden;display:none;" >Atualizado com sucesso</span></center>
        <span></span>
    </div>
</div>
<div id="conteudo_caixa">
    <div id="caixa_titulo"><span>Controle de Mesas</span></div>
    <div id="caixa_mesas">
        <div id="mesa1" class="mesaOn">
            <p><a href="#"
onclick="mostrarBalao(this,1);"><?=urldecode("Garçom")?></a></p>
            <p></p>
            <p><span>Fechar Conta</span></p>
        </div>
        <div id="mesa2" class="mesaOn">
            <p><a href="#"
onclick="mostrarBalao(this,2);">Garçom</a></p>

```

```

                                <p></p>

                                <p><span>Fechar Conta</span></p>
                                </div>
                                <div id="mesa3" class="mesaOn">
                                    <p><a                                href="#"
onclick="mostrarBalao(this,3);">Garçom</a></p>
                                    <p></p>
                                    <p><span>Fechar Conta</span></p>
                                </div>
                                <div id="mesa4" class="mesaOn">
                                    <p><a                                href="#"
onclick="mostrarBalao(this,4);">Garçom</a></p>
                                    <p></p>
                                    <p><span>Fechar Conta</span></p>
                                </div>
                                </div>
                                </div>
                                <script type="text/javascript">
                                    verificarStatus();
                                </script>
                                <?include($root."/template/footer.php");?>

```

## POCKET\_CLIENTE.PHP

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include("seguranca_cliente.php"); ?>
<?include($root."/template/header_pocket.php");?>
<div id="produtos">
    <?include($root."/modulos/produto/listaProdutos.php");?>
</div>
<?include($root."/template/footer_pocket.php");?>

```

## POCKET\_CONFIRMAR.PHP

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include("seguranca_cliente.php"); ?>
<?include($root."/template/header_pocket.php");?>

```

```

<div id="contas">
  <?include($root."/modulos/pedido/pocket_confirmar_pedido.php");?>
</div>
<?include($root."/template/footer_pocket.php");?>

```

### POCKET\_CONTA.PHP

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include("seguranca_cliente.php"); ?>
<?include($root."/template/header_pocket.php");?>
<div id="contas">
  <?include($root."/modulos/pedido/pocket_listaPedidos.php");?>
</div>
<?include($root."/template/footer_pocket.php");?>

```

### POCKET\_FECHAR\_CONTA.PHP

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include("seguranca_cliente.php"); ?>
<?include($root."/template/header_pocket.php");?>
<div class="padrao">
  <?include($root."/modulos/conta/pocket_fechar_conta.php");?>
</div>
<?include($root."/template/footer_pocket.php");?>

```

### POCKET\_LIBERAR\_MESA.PHP

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include("seguranca_cliente.php"); ?>
<?include($root."/template/header_pocket.php");?>
<div class="padrao">
  <? if(isset($_GET["erro"])){ ?>
    <div class="padrao">
      <span>Por favor informe a senha e a opção</span>
      <span><input          class="botoes"      type="button"      value="Voltar"
onclick="history.go(-1);" /></span>
    </div>
  <? } ?>
</div>
<?include($root."/template/footer_pocket.php");?>

```

## POCKET\_LOGIN.PHP

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include($root."/template/header_pocket.php");?>
<div id="loginPocket">
    <form action="controlador.php?action=getCliente" method="POST">
        <input type="hidden" name="idMesa" value="<?=$_GET["idMesa"]?>"/>
        <? if(!isset($_GET["passo1"])) { ?>
            <p>Informe seu telefone sem o DDD.</p>
            <p>Telefone:      <br/><input          type="text"          maxlength="8"
name="telCliente"></p>
            <? } else{ ?>
                <input          type="hidden"          name="telCliente"
value="<?=$_GET["telefone"]?>"/>
                <p>Telefone:      <br/><input          type="text"          maxlength="8"
name="telCliente" disabled="true" value="<?=$_GET["telefone"]?>"></p>
                <? if($_GET["passo1"] == "ok") { ?>
                    <input type="hidden" name="passo2" value="novo"/>
                    <p>Informe os dados restantes</p>
                    <p>Nome:      <br/><input          type="text"
name="nomeCliente"></p>
                    <p>Data Nascimento (ex. 10/01/2006): <br/>
                        <input  type="text"  class="dataNascimento"
name="nascimentoCliente[]" size="2" maxlength="2">/
                        <input  type="text"  class="dataNascimento"
name="nascimentoCliente[]" size="2" maxlength="2">/
                        <input  type="text"  class="dataNascimento"
name="nascimentoCliente[]" size="4" maxlength="4">
                    </p>
                    <? } else { ?>
                        <input          type="hidden"          name="passo2"
value="atualizar"/>
                        <input          type="hidden"          name="idCliente"
value="<?=$_GET["idCliente"]?>"/>
                        <p><b>Confirme</b> ou <b>altere</b> os dados
abaixo</p>
                        <p>Nome:      <br/><input          type="text"
name="nomeCliente" value="<?=$_GET["nome"]?>"></p>
                        <p>Data Nascimento (ex. 10/01/2006): <br/>

```

```

        <? $dataNascimento = explode ("/",
$_GET["nascimento"]);?>

        <input type="text" class="dataNascimento"
name="nascimentoCliente[]" size="2" maxlength="2" value="<?=$dataNascimento[0]?>">/
        <input type="text" class="dataNascimento"
name="nascimentoCliente[]" size="2" maxlength="2" value="<?=$dataNascimento[1]?>">/
        <input type="text" class="dataNascimento"
name="nascimentoCliente[]" size="4" maxlength="4" value="<?=$dataNascimento[2]?>">
        </p>
        <? } ?>
    <? } ?>
        <p><input type="submit" value="Enviar"/>
    <p></p>
</form>
</div>
<?
//Limpa a sessão
session_start("siscozinha");
session_unregister("conta_id");
session_unregister("nome_cliente");
session_destroy();
?>
<?include($root."/template/footer_pocket.php");?>

```

## POCKET\_LOGIN.PHP

```

<?require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php"?>
<?include("seguranca_cliente.php"); ?>
<?include($root."/template/header_pocket.php");?>
<div id="sucessoPedido">
    <span>Pedido Realizado com sucesso</span>
    <span><input class="botoes" type="button" value="Voltar"
onclick="window.location='pocket_cliente.php' /></span>
</div>
<?include($root."/template/footer_pocket.php");?>

```

## SEGURANCA.PHP

```

<?php

```

#Script para não permitir que o usuário digite uma URL do sistema e o acesse sem se autenticar

```
session_start();

if(isset($_SESSION['login'])){
    $loginAutenticado = $_SESSION['login'];
}
if (empty($loginAutenticado)){
    header("Location: index.php?msg=erro");
    exit;
}
$perfil = $_SESSION['perfil'];
?>
```

## **SEGURANCA\_CLIENTE.PHP**

```
<?php
```

#Script para não permitir que o usuário digite uma URL do sistema e o acesse sem se autenticar

```
session_start();
if(isset($_SESSION['conta_id'])){
    $idConta = $_SESSION['conta_id'];
    $nomeCliente = $_SESSION['nome_cliente'];
    $nomeGarcom = $_SESSION['nome_garcom'];
    $numeroMesa = $_SESSION['numeroMesa'];
}
if (empty($idConta)){
    header("Location: pocket_index.php");
    exit;
}
?>
```

## **CLIENTE.PHP**

```
<?php
```

```
/*
```

```
* Criado em 19/07/2006
```

```
* Autor(a): Mannuela Cruz
```

```
* Classe: Classe espelho da tabela Cliente
```

```
*/
```



```

class Cliente {
    var $id;
    var $telefone;
    var $nome;
    var $nascimento;

    function setId($id){
        $this->id = $id;
    }
    function setTelefone($telefone){
        $this->telefone = $telefone;
    }
    function setNome($nome){
        $this->nome = $nome;
    }
    function setNascimento($nascimento){
        $this->nascimento = $nascimento;
    }
    function getId(){
        return $this->id;
    }
    function getTelefone(){
        return $this->telefone;
    }
    function getNome(){
        return $this->nome;
    }
    function getNascimento(){
        return $this->nascimento;
    }
}
?>

```

## CONEXAO.PHP

```
<?
```

```
/**
```

```
 * Criado em 19/07/2006
```

```
 * Autor(a): Mannuela Cruz
```

```
 * Classe responsável por conectar ao banco de dado
```

```

*/
class Conexao {
    var $servidor = "localhost";
    var $usuario = "root";
    var $senha = "";
    var $banco = "mannu";
    var $conexao;

    function abrirConexao(){
        $this->conexao = mysql_connect($this->servidor,$this->usuario,$this->senha);
        if(!$this->conexao) {
            echo "Não foi possível conectar. Erro: ".mysql_error();
        } else {
            if(!mysql_select_db($this->banco,$this->conexao)){
                echo "Não foi possível selecionar o banco. Erro: ".mysql_error();
                return false;
            } else {
                return true;
            }
        }
    }

    function getConexao(){
        if($this->conexao == null){
            echo "Eh nulo";
            $this->abrirConexao();
        }
        return $this->conexao;
    }

    function close() {
        mysql_close($this->conexao);
    }

    function rollback(){
        mysql_query("ROLLBACK");
    }

    function msgErro($txt){

    }

}

?>

```

## CONTA.PHP

```

<?php
/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Classe: Classe espelho da tabela Conta
 */
class Conta {
    var $id;
    var $clienteld;
    var $mesald;
    var $datalni;
    var $dataFim;
    var $status;

    function setId($id){
        $this->id = $id;
    }
    function setClienteld($clienteld){
        $this->clienteld = $clienteld ;
    }
    function setMesald($mesald){
        $this->mesald = $mesald ;
    }
    function setDataIni($datalni){
        $this->datalni = $datalni ;
    }
    function setDataFim($dataFim){
        $this->dataFim = $dataFim ;
    }
    function setStatus($status){
        $this->status = $status ;
    }
    function getId(){
        return $this->id;
    }
    function getClienteld(){
        return $this->clienteld;
    }
}

```

```

        function getMesald(){
            return $this->mesald;
        }
        function getDataIni(){
            return $this->dataIni;
        }
        function getDataFim(){
            return $this->dataFim;
        }
        function getStatus(){
            return $this->status;
        }
    }
?>

```

## CONTROLADORCLIENTE.PHP

```

<?php
/*
 * Criado em 09/08/2006
 * Autor(a): Mannuela Cruz
 * Classe responsável pelas operações com o cliente
 */
class ControladorCliente{
    var $nomeTabela = "Cliente";
    var $colunaId = "Cli_ID";
    var $colunaTelefone = "Cli_Telefone";
    var $colunaNome = "Cli_Nome";
    var $colunaNascimento = "Cli_Dt_Nascimento";

    function getCliente($id){
        $conexao = new Conexao();
        if($conexao->abrirConexao()){
            $query = "select ".$this->colunaId.", ".
                $this->colunaTelefone.", ".
                $this->colunaNome.", ".
                $this->colunaNascimento." " .
                " from ".$this->nomeTabela." " .
                " where ".$this->colunaId." = ".$id;

```

```

$resultado = mysql_query($query,$conexao->getConexao());
if(mysql_num_rows($resultado) > 0){
    $linha = mysql_fetch_array($resultado);

    $cliente = new Cliente();
    $cliente->setId($linha[$this->colunaId]);
    $cliente->setTelefone($linha[$this->colunaTelefone]);
    $cliente->setNome($linha[$this->colunaNome]);
    $cliente->setNascimento($linha[$this->colunaNascimento]);
    $conexao->close();
    return $cliente;
}
$conexao->close();
return null;
}
}

function getClientePorTelefone($cliente){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $util = new Util();
        $query = "select ".$this->colunaId.", ".
                    $this->colunaTelefone.", ".
                    $this->colunaNome.", ".
                    $this->colunaNascimento." " .
                    " from ".$this->nomeTabela." " .
                    " where ".$this->colunaTelefone."='".$cliente-
>getTelefone()."";

        //" and ".$this->colunaNome."='".$cliente->getNome()." " .
        //" and ".$this->colunaNascimento."='".$util-
>transformarData($cliente->getNascimento()."";

        $resultado = mysql_query($query,$conexao->getConexao());
        if(mysql_num_rows($resultado) > 0){
            $linha = mysql_fetch_array($resultado);

            $cliente = new Cliente();
            $cliente->setId($linha[$this->colunaId]);
            $cliente->setTelefone($linha[$this->colunaTelefone]);

```

```

        $cliente->setNome($linha[$this->colunaNome]);
        $cliente->setNascimento($linha[$this->colunaNascimento]);
        $conexao->close();
        return $cliente;
    }
    $conexao->close();
    return null;
}
}

function incluirCliente($cliente){
    $conexao = new Conexao();
    $util = new Util();
    if($conexao->abrirConexao()){
        $query = "insert into ".$this->nomeTabela.
                    "(".$this->colunaTelefone.", ".
                    $this->colunaNome.", ".
                    $this->colunaNascimento.")" .
                    " values (".$cliente->getTelefone()."" .
                    ", ".$cliente->getNome()."" .
                    ", ".$util->transformarData($cliente->getNascimento())."");
        $resultado = mysql_query($query,$conexao->getConexao()) or die("Erro ao
incluir: ".mysql_error());
        $cliente->setId(mysql_insert_id());
        return $cliente;
    }
    $conexao->close();
    return false;
}

function atualizarCliente($cliente){
    $conexao = new Conexao();
    $util = new Util();
    if($conexao->abrirConexao()){
        $query = "update ".$this->nomeTabela.
                    " set ".$this->colunaNome."="."$cliente-
>getNome().", ".
                    $this->colunaNascimento."="."$util-
>transformarData($cliente->getNascimento())."" .
                    " where ".$this->colunaId."="."$cliente->getId()";

```

```

        $resultado = mysql_query($query,$conexao->getConexao()) or die("Erro ao
atualizar: ".mysql_error());
        return $cliente;
    }
    $conexao->close();
    return false;
}
}

?>

```

## CONTROLADORCONTA.PHP

```

<?php
/*
 * Criado em 09/08/2006
 * Autor(a): Mannuela Cruz
 * Classe responsável pelas operações com a conta
 */
class ControladorConta{
    var $nomeTabela = "Conta";
    var $colunaId = "Cta_ID";
    var $colunaIdCliente = "Cli_ID";
    var $colunaIdMesa = "Mesa_ID";
    var $colunaDataIni = "Cta_Data_Ini";
    var $colunaDataFim = "Cta_Data_Final";
    var $colunaStatus = "Cta_Status";

    function recuperarContaAtiva($mesa){
        $conexao = new Conexao();
        if($conexao->abrirConexao()){
            $idMesa = $mesa->getId();

            $query = "select ".$this->colunaId. " , " .
                $this->colunaIdCliente. " , " .
                $this->colunaIdMesa. " , " .
                $this->colunaDataIni. " , " .
                $this->colunaDataFim. " , " .
                $this->colunaStatus.
                " from " . $this->nomeTabela .

```

```

        " where ".$this->colunaIdMesa." = ".$idMesa." and ".$this->colunaStatus." <> 'F' ".

        " and DATE_FORMAT(".$this->colunaDataIni.','d/%m/%Y')

= DATE_FORMAT(now(),'d/%m/%Y')";

$resultado = mysql_query($query,$conexao->getConexao());
if(mysql_num_rows($resultado) > 0){
    $linha = mysql_fetch_array($resultado);

    $conta = new Conta();
    $conta->setId($linha[$this->colunaId]);
    $conta->setClientId($linha[$this->colunaIdCliente]);
    $conta->setMesId($linha[$this->colunaIdMesa]);
    $conta->setDataIni($linha[$this->colunaDataIni]);
    $conta->setDataFim($linha[$this->colunaDataFim]);
    $conta->setStatus($linha[$this->colunaStatus]);
    $conexao->close();
    return $conta;
}
return null;
}

}

function getConta($idConta){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $query = "select ".$this->colunaId. " , " .

            $this->colunaIdCliente. " , " .

            $this->colunaIdMesa. " , " .

            $this->colunaDataIni. " , " .

            $this->colunaDataFim. " , " .

            $this->colunaStatus.

            " from " . $this->nomeTabela .

            " where ".$this->colunaId." = ".$idConta;

        $resultado = mysql_query($query,$conexao->getConexao());
        if(mysql_num_rows($resultado) > 0){
            $linha = mysql_fetch_array($resultado);

            $conta = new Conta();

```



```

        $conta->setId($linha[$this->colunaId]);
        $conta->setClientId($linha[$this->colunaIdCliente]);
        $conta->setMesId($linha[$this->colunaIdMesa]);
        $conta->setDataIni($linha[$this->colunaDataIni]);
        $conta->setDataFim($linha[$this->colunaDataFim]);
        $conta->setStatus($linha[$this->colunaStatus]);
        $conexao->close();
        return $conta;
    }
    return null;
}

function recuperarContas($mesa){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $idMesa = $mesa->getId();

        $query = "select ".$this->colunaId. " , " .
                    $this->colunaIdCliente. " , " .
                    $this->colunaIdMesa. " , " .
                    $this->colunaDataIni. " , " .
                    $this->colunaDataFim. " , " .
                    $this->colunaStatus.
                    " from " . $this->nomeTabela .
                    " where ".$this->colunaIdMesa." = ".$idMesa.
                    " and DATE_FORMAT('".$this->colunaDataIni."','".$d/%m/%Y')
= DATE_FORMAT(now(),'d/%m/%Y')";
                    " and (".$this->colunaDataFim." is null or ".$this->colunaDataFim." = 0000000000000000)";

        $resultado = mysql_query($query,$conexao->getConexao());
        if(mysql_num_rows($resultado) == 1){
            $linha = mysql_fetch_array($resultado);

            $conta = new Conta();
            $conta->setId($linha[$this->colunaId]);
            $conta->setClientId($linha[$this->colunaIdCliente]);
            $conta->setMesId($linha[$this->colunaIdMesa]);

```

```

        $conta->setDataIni($linha[$this->colunaDataIni]);
        $conta->setDataFim($linha[$this->colunaDataFim]);
        $conta->setStatus($linha[$this->colunaStatus]);
        $conexao->close();
        return $conta;
    }else{
        while($linha = mysql_fetch_array($resultado)){
            if($linha[$this->colunaStatus] != "F"){
                $conta = new Conta();
                $conta->setId($linha[$this->colunaId]);
                $conta->setClientId($linha[$this->colunaIdCliente]);
                $conta->setMesId($linha[$this->colunaIdMesa]);
                $conta->setDataIni($linha[$this->colunaDataIni]);
                $conta->setDataFim($linha[$this->colunaDataFim]);
                $conta->setStatus($linha[$this->colunaStatus]);
                $conexao->close();
                return $conta;
            }
        }
    }
    return null;
}

function getHistoricoData($mesa,$data){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $idMesa = $mesa->getId();
        $listaConta = array();
        $query = "select ".$this->colunaId. " , " .
            $this->colunaIdCliente. " , " .
            $this->colunaIdMesa. " , " .
            $this->colunaDataIni. " , " .
            $this->colunaDataFim. " , " .
            $this->colunaStatus.
            " from " . $this->nomeTabela .
            " where ".$this->colunaIdMesa." = ".$idMesa." and ".$this->colunaStatus." = 'F' " .
            " and DATE_FORMAT('".$this->colunaDataIni."','".$d/%m/%Y')
            = DATE_FORMAT(now(),'".$data."')";
    }
}

```

```

$resultado = mysql_query($query,$conexao->getConexao());
while($linha = mysql_fetch_array($resultado)){
    $conta = new Conta();
    $conta->setId($linha[$this->colunaId]);
    $conta->setClientId($linha[$this->colunaIdCliente]);
    $conta->setMesId($linha[$this->colunaIdMesa]);
    $conta->setDataIni($linha[$this->colunaDataIni]);
    $conta->setDataFim($linha[$this->colunaDataFim]);
    $conta->setStatus($linha[$this->colunaStatus]);
    array_push($listaConta,$conta);
}
$conexao->close();
return $listaConta;
}

}

function reabrirConta($conta){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $idConta = $conta->getId();
        $query = "update " . $this->nomeTabela.
            " set ".$this->colunaStatus." = 'A'".
            " , ".$this->colunaDataFim." = 0000000000000000".
            " where ".$this->colunaId. "=".$idConta;

        $resultado = mysql_query($query,$conexao->getConexao());
        $resposta = mysql_affected_rows();
        $conexao->close();
        return $resposta;
    }
}

function fecharConta($conta){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $idConta = $conta->getId();
        $query = "update " . $this->nomeTabela.
            " set ".$this->colunaDataFim." = now()".
            " where ".$this->colunaId. "=".$idConta;

        $resultado = mysql_query($query,$conexao->getConexao());
        $resposta = mysql_affected_rows();
    }
}

```

```

        $conexao->close();
        return $resposta;
    }
}

function solicitarFechamento($conta){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $idConta = $conta->getId();
        $query = "update " . $this->nomeTabela.
            " set ".$this->colunaStatus." = 'S'".
            " where ".$this->colunaId. "=".$idConta;

        $resultado = mysql_query($query,$conexao->getConexao());
        $resposta = mysql_affected_rows();
        $conexao->close();
        return $resposta;
    }
}

function encerrarContaLiberarMesa($conta){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $idConta = $conta->getId();
        $query = "update " . $this->nomeTabela.
            " set ".$this->colunaStatus." = 'F'".
            " where ".$this->colunaId. "=".$idConta;

        $resultado = mysql_query($query,$conexao->getConexao());
        $resposta = mysql_affected_rows();
        $conexao->close();
        return $resposta;
    }
}

function incluirConta($conta){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $idConta = $conta->getId();
        $query = "insert into " . $this->nomeTabela.
            " (".$this->colunaIdCliente.", ".$this->colunaIdMesa.", ".$this->colunaDataIni.", ".$this->colunaStatus.")".
            " values (".$conta->getClientId().", ".$conta->getMesId().", now(), 'A')";
    }
}

```

```

        //echo $query;

        $resultado = mysql_query($query,$conexao->getConexao());
        $resposta = mysql_affected_rows();
        if($resposta != 0){
            $conta = $this->getConta(mysql_insert_id());
        }
        $conexao->close();
        return $conta;
    }
}
?>

```

## CONTROLADORGARCOM.PHP

```

<?php
/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Classe responsável pelas operações com garçons
 */

require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Garcom.php");
class ControladorGarcom{
    var $nomeTabela = "Garcom";
    var $colunaId = "Gar_ID";
    var $colunaNome = "Gar_Nome";

    function getListaGarcons(){
        $conexao = new Conexao();
        $listaGarcom = array();
        if($conexao->abrirConexao()){
            $query = "select ".$this->colunaId.", ".$this->colunaNome." from ".$this->nomeTabela."";

            $resultado = mysql_query($query,$conexao->getConexao());
            while($linha = mysql_fetch_array($resultado)){
                $garcom = new Garcom();
                $garcom->setId($linha[$this->colunaId]);
                $garcom->setNome($linha[$this->colunaNome]);
                array_push($listaGarcom,$garcom);
            }
        }
    }
}

```

```

    }
    $conexao->close();
    return $listaGarcom;
}
}
function getGarcom($id){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $query = "select ".$this->colunaId.", ".$this->colunaNome." from ".$this->nomeTabela." .
                " where ".$this->colunaId." = ".$id;

        $resultado = mysql_query($query,$conexao->getConexao());
        if(mysql_num_rows($resultado) > 0){
            $garcom = new Garcom();
            $linha = mysql_fetch_array($resultado);
            $garcom->setId($linha[$this->colunaId]);
            $garcom->setNome($linha[$this->colunaNome]);
            $conexao->close();
            return $garcom;
        }
        $conexao->close();
        return null;
    }
}
}
?>

```

## CONTROLADORMESA.PHP

```

<?php
/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Classe responsável pelas operações com mesas
 */
//require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Conexao.php");
//require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Mesa.php");
class ControladorMesa{

```

```

var $nomeTabela = "Mesa";
var $colunaIdGarcom = "Garcom_Gar_ID";
var $colunaId = "Mesa_ID";
var $colunaNome = "Mesa_Nome";
var $colunaData = "Mesa_Dt_escolhida";

function getListaMesas(){
    $conexao = new Conexao();
    $listaMesa = array();
    if($conexao->abrirConexao()){
        $query      =      "select      ".$this->colunaId.", ".$this->colunaNome.", ".$this->colunaIdGarcom." from ".$this->nomeTabela."";
        $resultado = mysql_query($query,$conexao->getConexao());
        while($linha = mysql_fetch_array($resultado)){
            $mesa = new Mesa();
            $mesa->setId($linha[$this->colunaId]);
            $mesa->setNome($linha[$this->colunaNome]);
            $mesa->setGarcomId($linha[$this->colunaIdGarcom]);
            array_push($listaMesa,$mesa);
        }
        $conexao->close();
        return $listaMesa;
    }
}

function getListaMesasNaoConfiguradas(){
    $conexao = new Conexao();
    $listaMesa = array();
    if($conexao->abrirConexao()){
        $query      =      "select      ".$this->colunaId.", ".$this->colunaNome.", ".$this->colunaIdGarcom." from ".$this->nomeTabela." .
                                " where ".$this->colunaData." <> now()";

        $resultado = mysql_query($query,$conexao->getConexao());
        while($linha = mysql_fetch_array($resultado)){
            $mesa = new Mesa();
            $mesa->setId($linha[$this->colunaId]);
            $mesa->setNome($linha[$this->colunaNome]);
            $mesa->setGarcomId($linha[$this->colunaIdGarcom]);
            array_push($listaMesa,$mesa);
        }
    }
}

```

```

    }
    $conexao->close();
    return $listaMesa;
}
}

function alterarMesas($listaMesas){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        for ($i = 0; $i < sizeof($listaMesas); $i++) {
            $mesa = $listaMesas[$i];
            $query = "update ".$this->nomeTabela." set ".$this->colunaIdGarcom."=".$mesa->getGarcomId()." ".
                "where ".$this->colunaId."=".$mesa->getId();
            $resultado = mysql_query($query,$conexao->getConexao());
        }
    }
    $conexao->close();
    return true;
}

function alterarMesa($mesa){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $query = "update ".$this->nomeTabela." set ".$this->colunaIdGarcom." =
            ".$mesa->getGarcomId()." ".
            "where ".$this->colunaId." = ".$mesa->getId();
        $resultado = mysql_query($query,$conexao->getConexao());
    }
    $conexao->close();
    return true;
}

function getMesa($id){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $query = "select ".$this->colunaId.", ".$this->colunaNome.", ".$this->colunaIdGarcom." from ".$this->nomeTabela.
            " where ".$this->colunaId." = ".$id;

        $resultado = mysql_query($query,$conexao->getConexao());
        if(mysql_num_rows($resultado) > 0){

```



```

        $linha = mysql_fetch_array($resultado);
        $mesa = new Mesa();
        $mesa->setId($linha[$this->colunaId]);
        $mesa->setNome($linha[$this->colunaNome]);
        $mesa->setGarcomId($linha[$this->colunaIdGarcom]);
        $conexao->close();
        return $mesa;
    }
    $conexao->close();
    return null;
}

function getMesaConfigurada($id){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $query = "select ".$this->colunaId.", ".$this->colunaNome.", ".$this->colunaIdGarcom." from ".$this->nomeTabela."
            where ".$this->colunaId." = ".$id." and ".$this->colunaData."= now()";

        $resultado = mysql_query($query,$conexao->getConexao());
        if(mysql_num_rows($resultado) > 0){
            $linha = mysql_fetch_array($resultado);
            $mesa = new Mesa();
            $mesa->setId($linha[$this->colunaId]);
            $mesa->setNome($linha[$this->colunaNome]);
            $mesa->setGarcomId($linha[$this->colunaIdGarcom]);
            $conexao->close();
            return $mesa;
        }
        $conexao->close();
        return null;
    }
}

function configurarMesa($id){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $query = "update ".$this->nomeTabela." set ".$this->colunaData." = now() " .
            "where ".$this->colunaId." = ".$id;
    }
}

```

```

        $resultado = mysql_query($query,$conexao->getConexao());
    }
    $conexao->close();
    return true;
}
}
?>

```

## CONTROLADORPEDIDO.PHP

```
<?php
```

```

/*
 * Criado em 11/08/2006
 * Autor(a): Mannuela Cruz
 * Classe responsável pelas operações com os pedidos
 */

class ControladorPedido {
    var $nomeTabela = "Pedido";
    var $colunaId = "Ped_ID";
    var $colunaIdConta = "Cta_ID";
    var $colunaEstado = "Ped_Estado";
    var $colunaDataReg = "Ped_Data_Reg";
    var $colunaDataFim = "Ped_Data_Fim";

    //Colunas Pedidos_has_Produto
    var $nomeTabelaRelacionamento = "Pedido_has_Produto";
    var $colunaHasIdProduto = "Prod_ID";
    var $colunaHasProdutoNome = "Prod_Nome";
    var $colunaHasProdutoPreco = "Prod_Preco";
    var $colunaHasQtde = "Has_Qtde";

    function incluirPedidoComProdutos($pedido) {
        $conexao = new Conexao();
        $util = new Util();
        if ($conexao->abrirConexao()) {
            mysql_query("BEGIN");

            $query = "insert into " . $this->nomeTabela .

```

```

        "(" . $this->colunaIdConta . " , " .
        $this->colunaEstado . " , " .
        $this->colunaDataReg . ") " .
        " values (" . $pedido->getContId() . "" .
        " , " . $pedido->getEstado() . "" .
        " , now());
        $resultado = mysql_query($query, $conexao->getConexao()) or die("Erro ao
incluir: " . mysql_error() . mysql_query("ROLLBACK"));
        $pedido->setId(mysql_insert_id());
        $listaProdutos = $pedido->getProdutos();
        for ($i = 0; $i < sizeof($listaProdutos); $i++) {
            $produto = $listaProdutos[$i];
            $query2 = "INSERT INTO " . $this->nomeTabelaRelacionamento . " " .
            "(" . $this->colunaId . " , " .
            $this->colunaHasIdProduto . " , " .
            $this->colunaHasQtde . ")" .
            " VALUES (" . $pedido->getId() . " , " . $produto->getId() . " , " .
            $produto->getQtdePedido() . ")";

            $resultado = mysql_query($query2, $conexao->getConexao()) or
die("Erro ao incluir pedidos: " . mysql_error() . mysql_query("ROLLBACK"));
        }
        mysql_query("COMMIT");
        $conexao->close();
        return $pedido;
    }
    $conexao->close();
    return false;
}

function statusPedido($mesa){
    $conexao = new Conexao();
    if ($conexao->abrirConexao()) {
        $idMesa = $mesa->getId();

        $query = "select min(ped.Ped_ID) as Ped_ID, mesa.Mesa_ID" .
            " from " . $this->nomeTabela . " ped, Conta cta, Mesa mesa,
pedido_has_produto has " .
            " where mesa.Mesa_ID = cta.Mesa_ID " .
            " and cta.Cta_ID = ped.Cta_ID " .

```

```

        " and ped.Ped_ID = has.Ped_ID " .
        " and Ped_Estado = 'A' " .
        " and cta.Cta_Status = 'A'" .
        " and DATE_FORMAT(ped.Ped_Data_Reg,'%d/%m/%Y') =
DATE_FORMAT(now(),'%d/%m/%Y')" .
        " and mesa.Mesa_ID = " . $idMesa.
        " group by mesa.Mesa_ID " .
        " order by ped.Ped_Data_Reg asc";

$resultado = mysql_query($query, $conexao->getConexao());
if(mysql_num_rows($resultado) == 1){
    $linha = mysql_fetch_array($resultado);
    $pedidoMesa = new PedidoMesa();
    $pedidoMesa->setPedido($this->getPedido($linha[$this->colunaId]));
    $controladorMesa = new ControladorMesa();
    $pedidoMesa->setMesa($controladorMesa-
>getMesa($linha["Mesa_ID"]));
    return $pedidoMesa;
}
}
return false;
}

//Retorna a fila de pedidos
function recuperarFila() {
    $conexao = new Conexao();
    $listaPedidoMesa = array();
    if ($conexao->abrirConexao()) {

        $query = "select min(ped.Ped_ID) as Ped_ID, mesa.Mesa_ID" .
        " from ". $this->nomeTabela ." ped, Conta cta, Mesa mesa,
pedido_has_produto has" .
        " where mesa.Mesa_ID = cta.Mesa_ID" .
        " and cta.Cta_ID = ped.Cta_ID" .
        " and ped.Ped_ID = has.Ped_ID" .
        " and Ped_Estado = 'A'" .
        " and cta.Cta_Status = 'A'" .
        " and DATE_FORMAT(ped.Ped_Data_Reg,'%d/%m/%Y') =
DATE_FORMAT(now(),'%d/%m/%Y')" .
        " group by mesa.Mesa_ID" .

```

```

" order by ped.Ped_Data_Reg asc";

$resultado = mysql_query($query, $conexao->getConexao());
while ($linha = mysql_fetch_array($resultado)) {
    $pedidoMesa = new PedidoMesa();
    $pedidoMesa->setPedido($this->getPedido($linha[$this->colunaId]));
    $controladorMesa = new ControladorMesa();
    $pedidoMesa->setMesa($controladorMesa-
>getMesa($linha["Mesa_ID"]));
    array_push($listaPedidoMesa,$pedidoMesa);
}
}
return $listaPedidoMesa;
}

function getPedido($id){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $query = "select ".$this->colunaId.", ".
                $this->colunaEstado.", ".
                $this->colunaDataReg.", ".
                $this->colunaDataFim.", ".
                $this->colunaIdConta.
                " from ".$this->nomeTabela.
                " where ".$this->colunaId." = ".$id;

        $resultado = mysql_query($query,$conexao->getConexao());
        if(mysql_num_rows($resultado) > 0){
            $linha = mysql_fetch_array($resultado);
            $pedido = new Pedido();
            $pedido->setId($linha[$this->colunaId]);
            $pedido->setEstado($linha[$this->colunaEstado]);
            $pedido->setDataReg($linha[$this->colunaDataReg]);
            $pedido->setDataFim($linha[$this->colunaDataFim]);
            $pedido->setContald($linha[$this->colunaIdConta]);
            $conexao->close();
            return $pedido;
        }
        $conexao->close();
    }
}

```

```

        return null;
    }

function getPedidoComProdutos($id){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $query = "select ".$this->colunaId.", ".
                    $this->colunaEstado.", ".
                    $this->colunaDataReg.", ".
                    $this->colunaDataFim.", ".
                    $this->colunaIdConta.
                    " from ".$this->nomeTabela.
                    " where ".$this->colunaId." = ".$id;

        $resultado = mysql_query($query,$conexao->getConexao());
        if(mysql_num_rows($resultado) > 0){
            $linha = mysql_fetch_array($resultado);
            $pedido = new Pedido();
            $pedido->setId($linha[$this->colunaId]);
            $pedido->setEstado($linha[$this->colunaEstado]);
            $pedido->setDataReg($linha[$this->colunaDataReg]);
            $pedido->setDataFim($linha[$this->colunaDataFim]);
            $pedido->setContId($linha[$this->colunaIdConta]);
            $conexao->close();
            return $pedido;
        }
        $conexao->close();
    }
    return null;
}

function encerrarPedido($pedido){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        mysql_query("BEGIN");
        $idPedido = $pedido->getId();
        $query = "update " . $this->nomeTabela.
                    " set ".$this->colunaDataFim." = now()".
                    " , ".$this->colunaEstado." = 'E'".
                    " where ".$this->colunaId." = ".$idPedido;
        $resultado = mysql_query($query,$conexao->getConexao())
    }
}

```

```

        or die("Erro ao atualizar pedidos: " . mysql_error() .
mysql_query("ROLLBACK"));

```

```

        $resposta = mysql_affected_rows();
        mysql_query("COMMIT");
        $conexao->close();
        return $resposta;
    }
}
?>

```

## CONTROLADORPRODUTO.PHP

```

<?php
/*
 * Criado em 09/08/2006
 * Autor(a): Mannuela Cruz
 * Classe responsável pelas operações com os produtos
 */
class ControladorProduto{
    var $nomeTabela = "Produto";
    var $colunaId = "Prod_ID";
    var $colunaNome = "Prod_Nome";
    var $colunaPreco = "Prod_Preco";
    var $colunaTempo = "Prod_Tempo_Preparo";
    var $colunaEstoque = "Prod_Estoque";
    var $colunaTipo = "Prod_Tipo";
    //Colunas Pedidos_has_Produto
    var $colunaHasQtde = "Has_Qtde";
    var $colunaHasAndamento = "Has_Andamento";

    function getProduto($id){
        $conexao = new Conexao();
        if($conexao->abrirConexao()){
            $query = "select ".$this->colunaId.", ".
                $this->colunaNome.", ".
                $this->colunaPreco.", ".
                $this->colunaTempo.", ".

```

```

        $this->colunaEstoque.", ".
        $this->colunaTipo." " .
        " from ".$this->nomeTabela."" .
        " where ".$this->colunaId." = ".$id;

$resultado = mysql_query($query,$conexao->getConexao());
if(mysql_num_rows($resultado) > 0){
    $linha = mysql_fetch_array($resultado);

    $produto = new Produto();
    $produto->setId($linha[$this->colunaId]);
    $produto->setNome($linha[$this->colunaNome]);
    $produto->setPreco($linha[$this->colunaPreco]);
    $produto->setTempo($linha[$this->colunaTempo]);
    $produto->setEstoque($linha[$this->colunaEstoque]);
    $produto->setTipo($linha[$this->colunaTipo]);

    $conexao->close();
    return $produto;
}
$conexao->close();
return null;
}
}

function getListaProdutos(){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $listaProdutos = array();

        $query = "select ".$this->colunaId.", ".
            $this->colunaNome.", ".
            $this->colunaPreco.", ".
            $this->colunaTempo.", ".
            $this->colunaEstoque.", ".
            $this->colunaTipo." " .
            " from ".$this->nomeTabela."" .
            " order by ".$this->colunaNome;

        $resultado = mysql_query($query,$conexao->getConexao());
    }
}

```



```

while($linha = mysql_fetch_array($resultado)){
    $produto = new Produto();
    $produto->setId($linha[$this->colunaId]);
    $produto->setNome($linha[$this->colunaNome]);
    $produto->setPreco($linha[$this->colunaPreco]);
    $produto->setTempo($linha[$this->colunaTempo]);
    $produto->setEstoque($linha[$this->colunaEstoque]);
    $produto->setTipo($linha[$this->colunaTipo]);
    array_push($listaProdutos,$produto);
}
$conexao->close();
return $listaProdutos;
}

function getProdutosPorPedido($idPedido){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $listaProdutos = array();

        $query = "select prod.".$this->colunaId.", " .
                "prod.".$this->colunaNome.", " .
                "prod.".$this->colunaPreco.", " .
                "php.".$this->colunaHasQtde.", " .
                "prod.".$this->colunaTipo.
                " from Pedido ped, Pedido_has_Produto php, Produto prod" .
                " where ped.Ped_ID = php.Ped_ID" .
                " and php.Prod_ID = prod.Prod_ID" .
                " and ped.Ped_ID =" . $idPedido;

        $resultado = mysql_query($query,$conexao->getConexao());

        while($linha = mysql_fetch_array($resultado)){
            $produto = new Produto();
            $produto->setId($linha[$this->colunaId]);
            $produto->setNome($linha[$this->colunaNome]);
            $produto->setPreco($linha[$this->colunaPreco]);
            $produto->setTipo($linha[$this->colunaTipo]);
            $produto->setQtdePedido($linha[$this->colunaHasQtde]);
        }
    }
}

```

```

        array_push($listaProdutos,$produto);
    }
    $conexao->close();
    return $listaProdutos;
}

}

function getProdutosPorConta($conta){
    $conexao = new Conexao();
    if($conexao->abrirConexao()){
        $idConta = $conta->getId();
        $listaProdutos = array();

        $query = "select prod.".$this->colunaNome.", " .
                "sum(php.Has_Qtde)          as          " . $this->
                ">colunaHasQtde.", " .
                "prod.".$this->colunaPreco.
        " from Conta conta, Pedido ped, Pedido_has_Produto php,
        Produto prod" .
        " where conta.Cta_ID = ped.Cta_ID" .
        " and ped.Ped_ID = php.Ped_ID" .
        " and php.Prod_ID = prod.Prod_ID" .
        " and conta.Cta_ID =" . $idConta.
        " group by prod.Prod_Nome";

        $resultado = mysql_query($query,$conexao->getConexao());
        while($linha = mysql_fetch_array($resultado)){
            $produto = new Produto();
            $produto->setNome($linha[$this->colunaNome]);
            $produto->setQtdePedido($linha[$this->colunaHasQtde]);
            $produto->setPreco($linha[$this->colunaPreco]);
            array_push($listaProdutos,$produto);
        }
        $conexao->close();
        return $listaProdutos;
    }
}
}

```

?>

## CONTROLADORUSUARIO.PHP

```
<?php
/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Classe responsável pelas operações com garçons
 */

class ControladorUsuario{
    var $nomeTabela = "Usuario";
    var $colunaId = "Usr_id";
    var $colunaLogin = "Usr_login";
    var $colunaSenha = "Usr_senha";
    var $colunaPerfil = "Usr_perfil";

    function getUsuario($objUsuario){
        $conexao = new Conexao();
        $login = $objUsuario->getLogin();
        $senha = $objUsuario->getSenha();

        if($conexao->abrirConexao()){
            $query = "select ".$this->colunaId.", ".$this->colunaLogin.", ".$this->colunaSenha.
                ", ".$this->colunaPerfil." from ".$this->nomeTabela." .
                " where ".$this->colunaLogin." = ".$login." AND ".$this->colunaSenha." = ".$senha."";

            $resultado = mysql_query($query,$conexao->getConexao());
            if(mysql_num_rows($resultado) > 0){
                $linha = mysql_fetch_array($resultado);
                $usuario = new Usuario();
                $usuario->setId($linha[$this->colunaId]);
                $usuario->setLogin($linha[$this->colunaLogin]);
                $usuario->setSenha($linha[$this->colunaSenha]);
                $usuario->setPerfil($linha[$this->colunaPerfil]);
                $conexao->close();
            }
        }
    }
}
```

```

        return $usuario;
    }
    $conexao->close();
    return null;
}
}
?>

```

## FACHADA.PHP

```

<?php
/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Arquivo que agrega todos os controladores e a classe de conexão
 * Para utilizar algum controlador basta que realize a inclusão deste
 */

require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Conexao.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/ControladorCliente.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/ControladorConta.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/ControladorMesa.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/ControladorGarcom.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/ControladorPedido.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/ControladorProduto.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/ControladorUsuario.php");

require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Cliente.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Conta.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Mesa.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/PedidoMesa.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Pedido.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Produto.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Usuario.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Util.php");
?>

```

## GARCOM.PHP

```

<?php

```

```

/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Classe: Produto espelho da tabela Garcom
 */
class Garcom {
    var $id;
    var $nome;

    function setId($id){
        $this->id = $id;
    }
    function setName($nome){
        $this->nome = $nome;
    }
    function getId(){
        return $this->id;
    }
    function getName(){
        return $this->nome;
    }
}
?>

```

## MESA.PHP

```

<?php
/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Classe: Classe espelho da tabela Mesa
 */
class Mesa {
    var $id;
    var $nome;
    var $garcomId;
    function setId($id){
        $this->id = $id;
    }
}

```

```

function setNome($nome){
    $this->nome = $nome;
}
function setGarcomId($garcomId){
    $this->garcomId = $garcomId;
}
function getId(){
    return $this->id;
}
function getNome(){
    return $this->nome;
}
function getGarcomId(){
    return $this->garcomId;
}
}
?>

```

## PEDIDO.PHP

```

<?php
/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Classe: Classe espelho da tabela Pedido
 */
class Pedido {
    var $id;
    var $contald;
    var $estado;
    var $dataReg;
    var $dataFim;
    var $produtos;

    function setId($id){
        $this->id = $id;
    }
    function setContald($contald){
        $this->contald = $contald;
    }
}

```

```

    }
    function setEstado($estado){
        $this->estado = $estado;
    }
    function setDataReg($dataReg){
        $this->dataReg = $dataReg;
    }
    function setDataFim($dataFim){
        $this->dataFim = $dataFim;
    }
    function setProdutos($produtos){
        $this->produtos = $produtos;
    }
    function getId(){
        return $this->id;
    }
    function getContald(){
        return $this->contald;
    }
    function getEstado(){
        return $this->estado;
    }
    function getDataReg(){
        return $this->dataReg;
    }
    function getDataFim(){
        return $this->dataFim;
    }
    function getProdutos(){
        return $this->produtos;
    }
    function addProduto($produto){
        if(!isset($this->produtos) || sizeof($this->produtos)==0){
            $this->produtos = array();
        }
        array_push($this->produtos,$produto);
    }
}
?>

```

**PEDIDOMESA.PHP**

```

<?php
/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Classe: Classe espelho da tabela PedidoMesa
 */
class PedidoMesa {
    var $pedido;
    var $mesa;
    var $status;

    function setPedido($pedido){
        $this->pedido = $pedido;
    }
    function setMesa($mesa){
        $this->mesa = $mesa;
    }
    function setStatus($status){
        $this->status = $status;
    }
    function getPedido(){
        return $this->pedido;
    }
    function getMesa(){
        return $this->mesa;
    }
    function getStatus(){
        return $this->status;
    }
}
?>

```

**PRODUTO.PHP**

```

<?php
/*
 * Criado em 19/07/2006

```



```

* Autor(a): Mannuela Cruz
* Classe: Produto espelho da tabela Produto
*/
class Produto{
    var $id ;
    var $nome;
    var $preco;
    var $tempo;
    var $estoque;
    var $tipo;
//Campos da tabela HAS
    var $qtdePedido;

    function setId($id){
        $this->id = $id;
    }
    function setNome($nome){
        $this->nome = $nome;
    }
    function setPreco($preco){
        $this->preco = $preco;
    }
    function setTempo($tempo){
        $this->tempo = $tempo;
    }
    function setEstoque($estoque){
        $this->estoque = $estoque;
    }
    function setTipo($tipo){
        $this->tipo = $tipo;
    }
    function setQtdePedido($qtdePedido){
        $this->qtdePedido = $qtdePedido;
    }
    function getId(){
        return $this->id;
    }
    function getNome(){

```

```

        return $this->nome;
    }
    function getPreco(){
        return $this->preco;
    }
    function getTempo(){
        return $this->tempo;
    }
    function getEstoque(){
        return $this->estoque;
    }
    function getTipo(){
        return $this->tipo;
    }
    function getQtdePedido(){
        return $this->qtdePedido;
    }
}
?>

```

## TESTESCONTROLADORES.PHP

```

<?
include "Fachada.php";
$pedido = new ControladorPedido();
echo "<pre>";
print_r($pedido->recuperarFila());
echo "</pre>";
?>

```

## USUARIO.PHP

```

<?php
/*
 * Criado em 26/07/2006
 * Autor(a): Mannuela Cruz
 * Classe: Usuario espelho da tabela Usuario
 */
class Usuario{
    var $id;

```

```

        var $login;
        var $senha;
        var $perfil;
        function setId($id){
            $this->id = $id;
        }
        function setLogin($login){
            $this->login = $login;
        }
        function setSenha($senha){
            $this->senha = $senha;
        }
        function setPerfil($perfil){
            $this->perfil = $perfil;
        }
        function getId(){
            return $this->id;
        }
        function getLogin(){
            return $this->login;
        }
        function getSenha(){
            return $this->senha;
        }
        function getPerfil(){
            return $this->perfil;
        }
    }
?>

```

## UTIL.PHP

```

<?php
/*
 * Criado em 22/08/2006
 *
 * Autor(a): Mannuela Cruz
 * Classe com alguns método auxiliares
 */
class Util {

```

```

function transformarData($data){
    if (strstr($data, "/")){
        $A = explode ("/", $data);
        $nova_data = $A[2] . "-". $A[1] . "-" . $A[0];
    }else{
        $A = explode("-", $data);
        $nova_data = $A[2] . "/" . $A[1] . "/" . $A[0];
    }
    return $nova_data;
}

function converteParaReal($valor){
    $valor = number_format($valor,2,"",".");
    return "R$ ".$valor;
}

}

}

?>

```

## POCKET\_GETCLIENTE.PHP

```

<?php
/*
 * Criado em 09/08/2006
 * Autor(a): Mannuela Cruz
 * Responsavel por recuperar um cliente, caso não existe inclui no banco de dados
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_POST["telCliente"])){
    echo "Informe todos os dados por favor";
    exit;
}
$mesaID = $_POST["idMesa"];
$cliente = new Cliente();
$cliente->setTelefone($_POST["telCliente"]);
$util = new Util();
$controladorCliente = new ControladorCliente();

if(isset($_POST["passo2"])){
    if($_POST["nomeCliente"]=="" || $_POST["telCliente"]=="" ||
$_POST["nascimentoCliente"] == ""){
        echo "informe todos os dados";
    }
}

```

```

        exit;
    }

    $cliente->setNome($_POST["nomeCliente"]);
    $cliente-
>setNascimento($_POST["nascimentoCliente"][0]."/".$_POST["nascimentoCliente"][1]."/".$_POST["na
scimentoCliente"][2]);

    $p2 = $_POST["passo2"];
    //Inclui ou altera um cliente e recupera o mesmo para juntamente com a mesa incluir
uma conta
    if($p2 == "novo"){
        $novoCliente = $controladorCliente->incluirCliente($cliente);
    }else{
        $cliente->setId($_POST["idCliente"]);
        $novoCliente = $controladorCliente->atualizarCliente($cliente);
    }
    $conta = new Conta();
    $conta->setMesald($mesaId);
    $conta->setClienteld($novoCliente->getId());
    $controladorConta = new ControladorConta();
    $conta = $controladorConta->incluirConta($conta);

    $controladorMesa = new ControladorMesa();
    $mesa = $controladorMesa->getMesa($conta->getMesald());

    $controladorGarcom = new ControladorGarcom();
    $garcom = $controladorGarcom->getGarcom($mesa->getGarcomId());

    session_start("siscozinha");
    session_name();
    session_destroy();
    session_register("conta_id");
    session_register("nome_cliente");
    session_register("nome_garcom");
    session_register("numeroMesa");

    $_SESSION['conta_id'] = $conta->getId();

```

```

$_SESSION['nome_cliente'] = $cliente->getNome();
$_SESSION['nome_garcom'] = $garcom->getNome();
$_SESSION['numeroMesa'] = $mesa->getNome();
header("Location: pocket_cliente.php");
}else {
    $novoCliente = $controladorCliente->getClientePorTelefone($cliente);
    if($novoCliente == null){
        header("Location:    pocket_login.php?idMesa=".$mesaID."&passo1=ok&telefone=".
$cliente->getTelefone());
    }else{
        header("Location:
pocket_login.php?idMesa=".$mesaID."&passo1=nao&idCliente=".$novoCliente-
>getId()."&telefone=".$novoCliente->getTelefone()."&nome=".$novoCliente-
>getNome()."&nascimento=". $util->transformarData($novoCliente->getNascimento()));
    }
}
?>

```

## FECHARCONTA.PHP

```

<?php
/*
 * Criado em 16/08/2006
 * Autor(a): Mannuela Cruz
 * Responsável por fechar a conta
 */

require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_GET["contaID"])){
    echo "Erro ao informar conta";
    exit;
}
$idConta = trim($_GET["contaID"]);
$controladorConta = new ControladorConta();
$conta = new Conta();
$conta->setId($idConta);
echo $controladorConta->fecharConta($conta);
?>

```

## LIBERARCONTAMESA.PHP

```
<?php
/*
 * Criado em 16/08/2006
 * Autor(a): Mannuela Cruz
 * Responsável por liberar a mesa encerrando a conta atual
 */

require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_GET["contaID"])){
    echo "Erro ao informar conta";
    exit;
}
$IdConta = trim($_GET["contaID"]);
$controladorConta = new ControladorConta();
$conta = new Conta();
$conta->setId($IdConta);
echo $controladorConta->encerrarContaLiberarMesa($conta);
?>
```

## POCKET\_FECHAR\_CONTA.PHP

```
<?php
/*
 * Criado em 29/08/2006
 *
 * Autor(a): Mannuela Cruz
 * Responsável por solicitar o fechamento da conta
 */

require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_POST["contaID"])){
    echo "Erro ao informar conta";
    exit;
}
$IdConta = $_POST["contaID"];
$controladorConta = new ControladorConta();
$conta = $controladorConta->getConta($IdConta);
$controladorConta->solicitarFechamento($conta);
$html = '<span>Aguarde em sua mesa.</span>';
```

```

$html .= '<span>Sua conta será encaminhada até sua mesa.</span>';
$html .= '<form action="controlador.php?action=pocketLiberarMesa" method="post">';
$html .= '<fieldset id="liberarMesa">';
$html .= '<legend>Acesso do Garçom</legend>';
$html .= '        <span>';
$html .= '            <select name="statusMesa">';
$html .= '                <option value="">Selecione</option>';
$html .= '                <option value="A">Re-abrir</option>';
$html .= '                <option value="F">Nova</option>';
$html .= '            </select>';
$html .= '        </span> ';
$html .= '        <span>Senha:<br/>';
$html .= '        <input type="password" name="senhaLiberarMesa" maxlength="10"/></span>
';

$html .= ' <input type="hidden" name="idConta" value="'. $idConta. '">';
$html .= '        <span><input type="submit" value="Confirmar"/>';
$html .= '</fieldset> ';
$html .= '</form>';
echo $html;
?>

```

## ALTERARGARCOM.PHP

```

<?php
/*
 * Criado em 01/08/2006
 * Autor(a): Mannuela Cruz
 *
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
$mesaID = $_GET["idMesa"];
$garcomID = $_GET["idGarcom"];
if(!isset($_GET["idMesa"]) || !isset($_GET["idGarcom"])){
    echo "Erro ao informar mesa e/ou garcom";
    exit;
}
$mesa = new Mesa();
$mesa->setId($mesaID);
$mesa->setGarcomId($garcomID);
$controladorMesa = new ControladorMesa();

```



```

echo $controladorMesa->alterarMesa($mesa);
?>

```

## COMBOGARCOM.PHP

```

<?php
/*
 * Criado em 01/08/2006
 * Autor(a): Mannuela Cruz
 * Retorna uma combo de acordo com a opção informada
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_GET["mesaID"])){
    echo "Erro ao informar mesa";
    exit;
}
$idMesa = trim($_GET["mesaID"]);
$controladorGarcom = new ControladorGarcom();
$controladorMesa = new ControladorMesa();
$mesa = $controladorMesa->getMesa($idMesa);
$listaGarcons = $controladorGarcom->getListaGarcons();

$html = '<div id="div_garcom">';
$html .= '<p>Escolha um garçom</p>';
$html .= '<form id="form_garcom" action="controlador.php?action=alterarGarcom">';
$html .= '<input type="hidden" name="idMesa" value="'. $mesa->getId().">';
$html .= '<p><select name="idGarcom">';
$html .= '
            <option value="">Selecione</option>';
            for ($j = 0; $j < sizeof($listaGarcons); $j++) {
                $garcom = $listaGarcons[$j];
                $html .= '
                    <option value="'. $garcom->getId().'"'. ($garcom->getId()==$mesa->getGarcomId()?'selected':'') .'>'. $garcom->getNome().'</option>';
            }
$html .= '
        </select></p>';
$html .= '<p><input type="button" value="Alterar" onclick="alterarGarcom();"></p>';
$html .= '</form>';
$html .= '</div>';

echo rawurlencode($html);

```

?>

## DETALHEMESA.PHP

```
<?php
/*
 * Criado em 09/08/2006
 * Autor(a): Mannuela Cruz
 * Retorna o detalhe de uma mesa
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_GET["mesaID"])){
    echo "Erro ao informar mesa";
    exit;
}
$IdMesa = trim($_GET["mesaID"]);
$controladorCliente = new ControladorCliente();
$controladorConta = new ControladorConta();
$controladorGarcom = new ControladorGarcom();
$controladorMesa = new ControladorMesa();

$mesa = $controladorMesa->getMesa($IdMesa);
$garcom = $controladorGarcom->getGarcom($mesa->getGarcomId());
$conta = $controladorConta->recuperarContaAtiva($mesa);
$historicoMesaAtual = $controladorConta->getHistoricoData($mesa,date("d/m/Y"));
$mesaVazia = false;
if($conta != null){
    $cliente = $controladorCliente->getCliente($conta->getClientId());
}else{
    $cliente = new Cliente();
    $cliente->setNome("Mesa vazia");
    $mesaVazia = true;
}
if($historicoMesaAtual == null){
    $msgSemHistorico = "Não existe histórico para o dia";
}

$html = '<div id="info_mesa_top">Caixa - '.$mesa->getNome().'</div>';
$html .= '<div id="info_mesa_conteudo">';
$html .= '<div id="info_mesa_dado_atual">';
```

```

$html .= '        <ul>';
$html .= '                <li>Conta Aberta';
$html .= '                        <ul>';
$html .= '                                <li>Cliente: <b>'.$cliente->getNome().</b></li>';
$html .= '                                <li>Garçom: <b>'.$garcom->getNome().</b></li>';
$html .= '                                </li>';
$html .= '                </li>';
if(!$mesaVazia){
    $html .= '                <li><input type="button" class="botaoOn" value="Conta Parcial"
onclick="carregarConta('.$conta->getId().')"></li>';
    $html .= '<input type="hidden" id="idContaAtual" value="'.$conta->getId().'">';
}
$html .= '        </ul>';
$html .= '</div>';
$html .= '<div id="info_mesa_dado_antigo">';
$html .= '        <ul>';
$html .= '                <li>Conta Anteriores';
$html .= '                        <ol>';
if(isset($msgSemHistorico)){
    $html .= '<li style="list-style-type:none;">'.$msgSemHistorico.</li>';
}else{
    for ($i = 0; $i < sizeof($historicoMesaAtual); $i++) {
        $contaTemp = $historicoMesaAtual[$i];
        $cliTemp = $controladorCliente->getCliente($contaTemp->getClientId());
        $html .= '<li><a href="#" onclick="carregarConta('.$contaTemp->getId().')">'.$cliTemp-
>getNome().</a></li>';
    }
}
$html .= '                </ol>';
$html .= '        </li>';
$html .= '    </ul>';
$html .= '</div>';
$html .= '<div id="info_mesa_rodape">';
if(!$mesaVazia){
    if($conta != null && ($conta->getDataFim() != "00000000000000" && $conta->getDataFim() !=
    "")){
        $html .= '                <span><input type="button" class="botaoOff" disabled="disabled"
id="fecharConta" value="Fechar Conta" onclick="fecharConta();"></span>';
    }else{

```

```

        $html .= '        <span><input    type="button"    class="botaoOn"    id="fecharConta"
value="Fechar Conta" onclick="fecharConta();"></span>';
    }
    $html .= '        <span><input type="button" class="botaoOn" id="liberarMesa" value="Liberar
Mesa" onclick="liberarMesa();"></span><br/><br/>';
    }
    $html .= '        <span><input        type="button"        class="botaoOn"        style="color:red"
onclick="fecharDetalheMesa();" value="Fechar"></span>';
    $html .= '</div>';
    $html .= '</div>';
    echo rawurlencode($html);
    //echo $html;
    ?>

```

## DETALHEMESABALCAO.PHP

```

<?php
/*
 * Criado em 09/08/2006
 * Autor(a): Mannuela Cruz
 * Retorna o detalhe de uma mesa
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_GET["mesaID"])){
    echo "Erro ao informar mesa";
    exit;
}
$IdMesa = trim($_GET["mesaID"]);
$controladorCliente = new ControladorCliente();
$controladorConta = new ControladorConta();
$controladorGarcom = new ControladorGarcom();
$controladorMesa = new ControladorMesa();
$controladorPedido = new ControladorPedido();

$mesa = $controladorMesa->getMesa($IdMesa);
$garcom = $controladorGarcom->getGarcom($mesa->getGarcomId());
$conta = $controladorConta->recuperarContaAtiva($mesa);

$pedidoMesa = $controladorPedido->statusPedido($mesa);

```

```

if($conta != null){
    $cliente = $controladorCliente->getCliente($conta->getClientId());
} else {
    $cliente = new Cliente();
    $cliente->setNome("Mesa vazia");
}
if($pedidoMesa == null){
    $msgSemPedido = "Não existe pedido ativo";
}
$html = '<div id="info_mesa_top">Balcão - '.$mesa->getNome().'/>';
$html .= '<div id="info_mesa_conteudo">';
$html .= '<div id="info_mesa_dado_atual">';
$html .= '        <ul>';
$html .= '            <li>Conta Aberta';
$html .= '        </li>';
$html .= '        <ul>';
$html .= '            <li>Cliente: <b>'.$cliente->getNome().'/b></li>';
$html .= '            <li>Garçom: <b>'.$garcom->getNome().'/b></li>';
$html .= '        </ul>';
$html .= '    </li>';
$html .= '</ul>';
$html .= '</div>';
$html .= '<div id="info_mesa_dado_antigo">';
$html .= '    <ul>';
$html .= '        <li>Pedidos do Cliente';
$html .= '    </li>';
if(isset($msgSemPedido)){
    $html .= '<li style="list-style-type:none;">'.$msgSemPedido.'</li>';
}else{
    $controladorProduto = new ControladorProduto();
    $pedido = $pedidoMesa->getPedido();
    $mesa = $pedidoMesa->getMesa();
    $listaProdutos = $controladorProduto->getProdutosPorPedido($pedido->getId());
    for ($i = 0; $i < sizeof($listaProdutos); $i++) {
        $produtoTemp = $listaProdutos[$i];
        $html .= '<li>'.$produtoTemp->getNome().'( <span style="color:black;font-weight:bold;">'.$produtoTemp->getQtdePedido().'/span> )</a></li>';
    }
}
}

```

```

$html .= '                                </ol>';
$html .= '                                </li>';
$html .= '                                </ul>';
$html .= '</div>';
$html .= '<div id="info_mesa_rodape">';
if(!isset($msgSemPedido)){
    $html .= '        <span><input        type="button"        class="botaoOn"        id="liberarPedido"
value="Liberar Pedido" onclick="liberarPedido();"></span><br/><br/>';
    $html .= '<input type="hidden" id="idPedidoAtual" value="'. $pedido->getId().">';
}
$html .= '        <span><input        type="button"        class="botaoOn"        style="color:red"
onclick="fecharDetalheMesa();" value="Fechar"></span>';
$html .= '</div>';
$html .= '</div>';
echo rawurlencode($html);
//echo $html;
?>

```

## ESCOLHERMESA.PHP

```

<?php
/*
 * Criado em 01/08/2006
 * Autor(a): Mannuela Cruz
 *
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
$mesaID = $_POST["idMesa"];
if(!isset($_POST["idMesa"])){
    echo "Erro ao informar mesa";
    exit;
}
$controladorMesa = new ControladorMesa();
$mesaAtiva = $controladorMesa->getMesaConfigurada($mesaID);
if($mesaAtiva == ""){
    $controladorMesa->configurarMesa($mesaID);
    header("Location: pocket_login.php?idMesa=".$mesaID);
}else{
    echo "Essa mesa já se encontra ocupada";
}

```

```
}
?>
```

## **GARCOMMESA.PHP**

```
<?php
/*
 * Criado em 02/08/2006
 * Autor(a): Mannuela Cruz
 * Retorna o nome do garçom de acordo com uma mesa
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_GET["mesaID"])){
    echo "Erro ao informar mesa";
    exit;
}
$idMesa = trim($_GET["mesaID"]);
$controladorMesa = new ControladorMesa();
$mesa = $controladorMesa->getMesa($idMesa);

$controladorGarcom = new ControladorGarcom();
$garcom = $controladorGarcom->getGarcom($mesa->getGarcomId());
echo rawurlencode($garcom->getNome());

?>
```

## **MESAS. PHP**

```
<?php
/*
 * Criado em 01/08/2006
 * Autor(a): Mannuela Cruz
 *
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
$controladorMesa = new ControladorMesa();
$listaMesas = $controladorMesa->getListaMesasNaoConfiguradas();
$html = '<select name="idMesa">';
for ($i = 0; $i < sizeof($listaMesas); $i++) {
    $mesa = $listaMesas[$i];
```

```

$html .= '<option value="'. $mesa->getId().">'. $mesa->getNome().'</option>';
}
$html .= '</select>';
echo $html;
?>

```

## POCKET\_LIBERAR\_MESA.PHP

```

<?php
/*
 * Criado em 29/08/2006
 *
 * Autor(a): Mannuela Cruz
 * Responsável por liberar ou reativar uma mesa
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
require($_SERVER["DOCUMENT_ROOT"]."/sistema/config_server.php");
if((!isset($_POST["senhaLiberarMesa"]) || !isset($_POST["statusMesa"])) ||
    ($_POST["senhaLiberarMesa"]=="" || $_POST["statusMesa"]=="")){
    header("Location: pocket_liberar_mesa.php?erro=ok");
    exit;
}
$senhaIntranet = $_POST["senhaLiberarMesa"];
$opcao = $_POST["statusMesa"];
//Validar senha aqui
if($senhaIntranet != $senhaGarcom){
    header("Location: pocket_liberar_mesa.php?erro=ok");
    exit;
}

$idConta = $_POST["idConta"];

$controladorConta = new ControladorConta();

$conta = $controladorConta->getConta($idConta);

if($opcao == "A"){
    $controladorConta->reabrirConta($conta);
    header("Location: pocket_cliente.php");
}else{

```



```

        header("Location: pocket_login.php?idMesa=".$conta->getMesald());
    }
?>

```

## STATUSMESAS.PHP

```

<?php
/*
 * Criado em 11/08/2006
 * Autor(a): Mannuela Cruz
 * Retorna o status de uma mesa
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");

$controladorConta = new ControladorConta();
$statusMesas = "";

for ($i = 0; $i < 4; $i++) {
    $mesa = new Mesa();
    $mesa->setId(($i+1));
    $conta = $controladorConta->recuperarContas($mesa);
    if($conta != null){
        switch($conta->getStatus()){
            case "S":$statusMesas .= "mesaPause";
            break;
            case "F":$statusMesas .= "mesaOn";
            break;
            case "A":$statusMesas .= "mesaOff";
        }
    }else{
        $statusMesas .= "mesaOn";
    }
    if($i+1 < 4){
        $statusMesas .= ",";
    }
}
echo $statusMesas;
?>

```

## LIBERAR PEDIDO MESA.PHP

```
<?php
/*
 * Criado em 16/08/2006
 * Autor(a): Mannuela Cruz
 * Responsável por liberar o pedido atual
 */

require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_GET["pedidoID"])){
    echo "Erro ao informar conta";
    exit;
}
$IdPedido = trim($_GET["pedidoID"]);
$controladorPedido = new ControladorPedido();
$pedido = new Conta();
$pedido->setId($IdPedido);
echo $controladorPedido->encerrarPedido($pedido);
?>
```

## LISTAR PEDIDOS.PHP

```
<?php
/*
 * Criado em 11/08/2006
 * Autor(a): Mannuela Cruz
 * Responsável por listar os produtos pedidos de uma determinada conta
 */

require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_GET["contaID"])){
    echo "Erro ao informar conta";
    exit;
}
$IdConta = trim($_GET["contaID"]);
$controladorProduto = new ControladorProduto();
$conta = new Conta();
$conta->setId($IdConta);

$util = new Util();
```

```

$listaProdutos = $controladorProduto->getProdutosPorConta($conta,true);
$subTotal = 0;
$servicos = 1.1;
$html = '<div id="tabelaContaCozinha">';
$html .= '        <table>';
$html .= '            <thead>';
$html .= '                <tr>';
$html .= '                    <td width="450">Itens</td>';
$html .= '                    <td>Quantidades</td>';
$html .= '                    <td>Preço Unitário</td>';
$html .= '                </tr>';
$html .= '            </thead>';
$html .= '            <tbody>';
                for ($i = 0; $i < sizeof($listaProdutos); $i++) {
                    $produto = $listaProdutos[$i];
                    $subTotal+=$(produto->getPreco()*produto->getQtdePedido());

$html .= '                <tr>';
$html .= '                    <td>'. $produto->getNome(). '</td>';
$html .= '                    <td>'. $produto->getQtdePedido(). '</td>';
$html .= '                    <td>'. $util->converteParaReal($produto->getPreco()). '</td>';
$html .= '                </tr>';
                }

$html .= '            </tbody>';
$html .= '            <tfoot>';
$html .= '                <tr>';
$html .= '                    <td colspan="2">SUB-TOTAL:</td>';
$html .= '                    <td>'. $util->converteParaReal($subTotal). '</td>';
$html .= '                </tr>';
$html .= '                <tr>';
$html .= '                    <td colspan="2">Serviços:</td>';
$html .= '                    <td>'. $util->converteParaReal($subTotal*0.1). '</td>';
$html .= '                </tr>';
$html .= '                <tr>';
$html .= '                    <td colspan="2">TOTAL:</td>';
$html .= '                    <td>'. $util->
>converteParaReal($subTotal*$servicos). '</td>';
$html .= '                </tr>';
$html .= '            </tfoot>';

```

```

$html .= '          </table>';
$html .= '</div>';
$html .= '<div id="botoes_listaPedidos">';
$html .= '          <input          type="button"          class="botaoOn"          value="Imprimir"
onclick="window.print();" />';
        $html .= '          <input type="button" class="botaoOn" value="Voltar" onclick="voltarConta();"
style="color:red" />';
$html .= '</div>';
echo rawurlencode($html);

```

## POCKET\_CONFIRMAR\_PEDIDO.PHP

```

<?php

/*
 * Criado em 24/08/2006
 *
 * Autor(a): Mannuela Cruz
 * Responsável por listar todos os produtos
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
if(!isset($_POST["produtoIDS"])){
    echo "Erro ao informar produtos";
    exit;
}
$produtos = explode(";", $_POST["produtoIDS"]);
$controladorProduto = new ControladorProduto();

$util = new Util();

$listaProdutos = array();
$listaProdQtde = "";
$subTotal = 0;
$isEmpty = true;
for($i=0;$i<sizeof($produtos);$i++){
    $qtde = $_POST["prod".$i];
    if($qtde != 0){
        $isEmpty = false;
        $produto = $controladorProduto->getProduto($produtos[$i]);
        $produto->setQtdePedido($qtde);
    }
}

```

```

        array_push($listaProdutos,$produto);
        $listaProdQtde .= $produto->getId().",".$qtde;
        if($i+1<sizeof($produtos)){
            $listaProdQtde.=",";
        }
    }
}

if($isVazio){
    $html = '<div class="padrao">';
    $html .= '<span>Informe ao menos um produto</span>';
    $html .= ' <span><input class="botoes" type="button" value="Voltar"
onclick="window.location=\'pocket_cliente.php\' /></span>';
    $html .= '</div>';
    echo $html;
    exit;
}

if(substr($listaProdQtde,-1,1) == ","){
    $listaProdQtde = substr($listaProdQtde,0,strlen($listaProdQtde)-1);
}

$html = '<div id="tabelaPedidos">';
$html .= '<form action="controlador.php?action=realizarPedido" method="post">';
$html .= ' <table width="260">';
$html .= ' <thead>';
$html .= ' <tr>';
$html .= ' <td width="50%">Itens</td>';
$html .= ' <td width="20%" class="dadosConta">Qtde</td>';
$html .= ' <td width="30%" class="dadosConta">Preço</td>';
$html .= ' </tr>';
$html .= ' </thead>';
$html .= ' <tbody>';

    for ($i = 0; $i < sizeof($listaProdutos); $i++) {
        $produto = $listaProdutos[$i];
        $subTotal+=( $produto->getPreco()*$produto->getQtdePedido());

        $html .= ' <tr>';
        $html .= ' <td>'. $produto->getNome().'</td>';
        $html .= ' <td class="dadosConta">'. $produto->getQtdePedido().'</td>';

```

```

        $html .= '                                <td class="dadosConta">'. $util->converteParaReal($produto-
>getPreco()).'</td>';
        $html .= '                                </tr>';
    }
    $html .= '                                </tbody>';
    $html .= '                                <tfoot>';
    $html .= '                                <tr>';
    $html .= '                                <td colspan="2" class="labelConta">SUB-
TOTAL:</td>';
    $html .= '                                <td class="dadosFinal">'. $util-
>converteParaReal($subTotal).'</td>';
    $html .= '                                </tr>';
    $html .= '                                </tfoot>';
    $html .= '                                </table>';
    $html .= '</div>';
    $html .= '<div style="text-align:center;">';
    $html .= '<input type="hidden" name="listaProdQtde" value="'. $listaProdQtde.'">';
    $html .= '<input type="hidden" name="idConta" value="'. $idConta.'">';
    $html .= '<input class="botoes" type="button" value="Voltar"
onclick="window.location=\'pocket_cliente.php\'" />&nbsp;';
    $html .= '<input class="botoes" type="submit" value="Confirmar"/>';
    $html .= '</div>';
    $html .= '</form>';
    echo $html;
?>

```

## POCKET\_LISTAPEDIDOS.PHP

```

<?php
/*
 * Criado em 28/08/2006
 *
 * Autor(a): Mannuela Cruz
 *
 */
require($_SERVER["DOCUMENT_ROOT"].'/sistema/libs/Fachada.php');
if(!isset($_GET["contaID"])){
    echo "Erro ao informar conta";
    exit;
}

```

```

$idConta = trim($_GET["contaID"]);
$controladorProduto = new ControladorProduto();

$conta = new Conta();
$conta->setId($idConta);

$util = new Util();

$listaProdutos = $controladorProduto->getProdutosPorConta($conta);
$subTotal = 0;
$servicos = 1.1;
$html = '<div id="tabelaPedidos">';
$html .= '        <table width="260">';
$html .= '            <thead>';
$html .= '                <tr>';
$html .= '                    <td width="50%">Itens</td>';
$html .= '                    <td width="20%" class="dadosConta">Qtde</td>';
$html .= '                    <td width="30%" class="dadosConta">Preço
Unitário</td>';
$html .= '                </tr>';
$html .= '            </thead>';
$html .= '            <tbody>';
                for ($i = 0; $i < sizeof($listaProdutos); $i++) {
                    $produto = $listaProdutos[$i];
                    $subTotal+=$( $produto->getPreco()*$produto->getQtdePedido());
                    $html .= '                <tr>';
                    $html .= '                    <td>'. $produto->getNome(). '</td>';
                    $html .= '                    <td class="dadosConta">'. $produto-
>getQtdePedido(). '</td>';
                    $html .= '                    <td class="dadosConta">'. $util->converteParaReal($produto-
>getPreco()). '</td>';
                    $html .= '                </tr>';
                }
            $html .= '            </tbody>';
            $html .= '            <tfoot>';
            $html .= '                <tr>';
            $html .= '                    <td colspan="2" class="labelConta">SUB-
TOTAL:</td>';

```

```

        $html .= '                                <td                                class="dadosFinal">'. $util-
>converteParaReal($subTotal). '</td>';
        $html .= '                                </tr>';
        $html .= '                                <tr>';
        $html .= '                                <td colspan="2" class="labelConta">Serviços:</td>';
        $html .= '                                <td                                class="dadosFinal">'. $util-
>converteParaReal($subTotal*0.1). '</td>';
        $html .= '                                </tr>';
        $html .= '                                <tr>';
        $html .= '                                <td colspan="2" class="labelConta">TOTAL:</td>';
        $html .= '                                <td                                class="dadosFinal">'. $util-
>converteParaReal($subTotal*$servicos). '</td>';
        $html .= '                                </tr>';
        $html .= '                                </tfoot>';
        $html .= '                                </table>';
        $html .= '</div>';
        $html .= '<div style="text-align:center;">';
        $html .= '<form action="pocket_fechar_conta.php" method="post">';
        $html .= '                                <input                                class="botoes"                                type="button"                                value="Voltar"
onclick="window.location=\'\pocket_cliente.php\'" />';
        $html .= '                                <input type="hidden" name="contalD" value="'. $idConta. '" /> ';
        $html .= '                                <input class="botoes" type="submit" value="Fechar Conta"/>';
        $html .= '                                </form>';
        $html .= '</div>';
        echo $html;
?>

```

## POCKET\_REALIZARPEDIDO.PHP

```

<?php
/*
 * Criado em 28/08/2006
 *
 * Autor(a): Mannuela Cruz
 *
 */
//
require ($_SERVER["DOCUMENT_ROOT"] . "/sistema/libs/Fachada.php");
if(!isset($_POST["listaProdQtdes"])){
    echo "Erro ao informar pedido";
}

```



```

        exit;
    }
    $controladorPedido = new ControladorPedido();
    $produtosQtde = explode(";", $_POST["listaProdQtdes"]);

    $pedido = new Pedido();
    $pedido->setContald($_POST["idConta"]);
    $pedido->setEstado("A");

    for($i=0;$i< sizeof($produtosQtde); $i++){
        $produto = new Produto();
        $tmpProduto = explode(";", $produtosQtde[$i]);
        $produto->setId($tmpProduto[0]);
        $produto->setQtdePedido($tmpProduto[1]);
        $pedido->addProduto($produto);
    }
    $novoPedido = $controladorPedido->incluirPedidoComProdutos($pedido);
    header("Location: pocket_pedido_confirmado.php");
?>

```

## STATUSPEDIDO.PHP

```

<?php
/*
 * Criado em 11/08/2006
 * Autor(a): Mannuela Cruz
 * Retorna o status de uma mesa
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");

$controladorPedido = new ControladorPedido();
$fila = $controladorPedido->recuperarFila();
$statusMesas = "";

for ($i = 0; $i < 4; $i++) {
    $mesa = new Mesa();
    $mesa->setId(($i+1));
    $pedidoMesa = $controladorPedido->statusPedido($mesa);
    if($pedidoMesa !== false){
        $controladorProduto = new ControladorProduto();
    }
}

```

```

$pedido = $pedidoMesa->getPedido();
$mesa = $pedidoMesa->getMesa();
$listaProdutos = $controladorProduto->getProdutosPorPedido($pedido->getId());
/*
$isProntoCozinha = false;
for ($j=0; $j< sizeof($listaProdutos); $j++) {
    $produto = $listaProdutos[$j];
    //Se existir um pedido pronto enviar alerta para o balcão
    if($produto->getTipo() == 0 && $produto->getStatus() == "P"){
        $isProntoCozinha = true;
    }
}*/

for($id=0;$id<sizeof($fila);$id++){
    $mesaTmp = $fila[$id]->getMesa();
    if($mesaTmp->getId() == $mesa->getId()){
        $ordem = $id+1;
    }
}

$cor = "mesaOff";
/*if($isProntoCozinha === true){
    $cor = "mesaPause";
} */

$statusMesas .= "mesa".$mesa->getId().".".$cor."=".$ordem;
//$statusMesas .= "mesa".$mesa->getId().".".$cor;
}else{
    $statusMesas .= "mesa".$mesa->getId()."=mesaOn=0";
    //$statusMesas .= "mesa".$mesa->getId()."=mesaOn=0";
}
if($i+1 < 4){
    $statusMesas .=",";
}
}
echo $statusMesas;
?>

```

## LISTAPRODUTOS.PHP

```
<?php
```

```

/*
 * Criado em 24/08/2006
 *
 * Autor(a): Mannuela Cruz
 * Responsável por listar todos os produtos
 */
require ($_SERVER["DOCUMENT_ROOT"] . "/sistema/libs/Fachada.php");
$controladorProduto = new ControladorProduto();
$listaProdutos = $controladorProduto->getListaProdutos();

$stabelaComida = $stabelaBebida = "<table>";
$stalids = "";
$qtdeComida = $qtdeBebida = 0;
for ($i = 0; $i < sizeof($listaProdutos); $i++) {
    $produto = $listaProdutos[$i];
    $stalids .= $produto->getId();
    if($i+1 < sizeof($listaProdutos)){
        $stalids .= ",";
    }
    $stabelaComida .= "<tr>";
    $stabelaBebida .= "<tr>";
    if ($produto->getTipo() == "0") {
        $stabelaComida .= '<td>' . $produto->getNome() . '</td><td>';
        $stabelaComida .= '<select name="prod" . $i . ">';
        for ($j = 0; $j <= 10; $j++) {
            $stabelaComida .= '<option value="" . $j . ">' . $j . '</option>';
        }
        $stabelaComida .= '</select></td>';
        $qtdeComida++;
    } else {
        $stabelaBebida .= '<td>' . $produto->getNome() . '</td><td>';
        $stabelaBebida .= '<select name="prod" . $i . ">';
        for ($j = 0; $j <= 10; $j++) {
            $stabelaBebida .= '<option value="" . $j . ">' . $j . '</option>';
        }
        $stabelaBebida .= '</select></td>';
        $qtdeBebida++;
    }
}

```

```

}

$tabelaComida .= "</tr>";
$tabelaBebida .= "</tr>";
$tabelaComida .= "</table>";
$tabelaBebida .= "</table>";

$html = '<form action="pocket_confirmar.php" method="post">';
$html .= '<input type="hidden" name="produtoIDS" value="'. $listaIds .'">';
$html .= '<table>';
$html .= '<tr><td valign="top">' . $tabelaComida . '</td><td valign="top">' . $tabelaBebida .
'</td></tr>';
$html .= '<tr><td style="text-align:center"><input class="botaoEnviar"
onclick="window.location=\'pocket_conta.php?contaID=' . $idConta . '\'' type="button" value="Conta
Parcial"></td>';
$html .= '<td style="text-align:center"><input class="botaoEnviar" type="submit"
value="Confirmar"></td></tr>';
$html .= '</table>';
$html .= '</form>';
echo $html;
?>

```

## AUTENTICACAO.PHP

```

<?php
/*
 * Criado em 19/07/2006
 * Autor(a): Mannuela Cruz
 * Arquivo responsável por autenticar um usuário interno do sistema
 */
require($_SERVER["DOCUMENT_ROOT"]."/sistema/libs/Fachada.php");
$login = trim($_GET["txtLogin"]);
$senha = trim($_GET["txtSenha"]);

#Encripta a senha com MD5
if(!empty($senha)){
    $senha = md5($senha);
}
$usuario = new Usuario();
$usuario->setLogin($login);

```

```

$usuario->setSenha($senha);
$controladorUsuario = new ControladorUsuario();

$usuarioAutenticado = $controladorUsuario->getUsuario($usuario);
if($usuarioAutenticado == null){
    echo "false";
}else{
    session_start("siscozinha");
    session_name();
session_destroy();
session_register("login");
session_register("perfil");
$_SESSION['login'] = $usuarioAutenticado->getLogin();
$_SESSION['perfil'] = $usuarioAutenticado->getPerfil();
echo $usuarioAutenticado->getPerfil();
}
?>

```

## FOOTER.PHP

```

        <? if(isset($loginAutenticado)){ ?>
            <div id="sair_sistema">
                <span><a href="<?=$documentRoot?>?msg=logout">Sair do
Sistema</a></span>
            </div>
        <? } ?>
    </div>
</div>
<div id="rodape">
    <div>
        <p>Sistema desenvolvido por Mannuela Cruz</p>
        <p>Todos os direitos reservados.</p>
    </div>
</div>
</body>
</html>

```

## FOOTER\_POCKET.PHP

```

</div>

```

```
</body>
</html>
```

## HEADER.PHP

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta http-equiv="pragma" content="no-cache" />
<title>Sistema</title>

<script type="text/javascript" src="<?=$documentRoot?>/js/AjaxRequest.js"></script>
<script type="text/javascript" src="<?=$documentRoot?>/js/prototype.js"></script>
<script type="text/javascript" src="<?=$documentRoot?>/js/scriptaculous.js"></script>
<script type="text/javascript" src="<?=$documentRoot?>/js/funcoes.js"></script>
<script type="text/javascript" src="<?=$documentRoot?>/js/overlib/overlib.js"></script>
<style type="text/css">
    @IMPORT url("<?=$documentRoot?>/css/estilos.css");
</style>
</head>
<body>
<div id="principal">
    <div id="cabecalho"></div>
    <div id="conteudo">
```

## HEADER\_POCKET.PHP

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta http-equiv="pragma" content="no-cache" />
<title>Sistema</title>

<style type="text/css">
```

```

@IMPORT url("<?=$documentRoot?>/css/palmEstilos.css");
</style>
</head>
<body>
<div id="principal">
<div id="cabecalho">
</div>
<? if(isset($nomeCliente) && isset($nomeGarcom)){?>
    <div id="dados_mesa">
        <div class="mesaGarcom"><?=$numeroMesa?>/<?=$nomeGarcom?></div>
        <div class="nomeCliente"><?=$nomeCliente?></div>
    </div>
<? } ?>

```

## CAIXA.CSS

```

#info_mesa {
border: 1px solid;
margin-top: 20px;
margin-left: 75px;
position: absolute;
width: 650px;
height: 350px;
background-color: #FFFFFF;
z-index: 1;
/*background-image: url("../img/dados_mesa_bg.png");*/
}

```

```

#info_mesa_top{
font-family: Verdana;
font-size: 14px;
font-weight: bold;
text-align: center;
color: #23B3B8;
height: 18px;
margin-bottom: 10px;
}
#info_mesa_dado_atual{
width: 325px;
float: left;

```

```

height: 270px;
}
#info_mesa_dado_atual ul{
font-size: 12px;
}
#info_mesa_dado_atual ul li{
list-style-type: none;
margin-bottom: 10px;
}
/* Informações dentro da conta */
#info_mesa_dado_atual ul li ul li{
list-style-type: none;
font-size: 10px;
margin-bottom: 2px;
color: #FF0000;
}
#info_mesa_dado_antigo{
background-color: #F9F9F9;
width: 325px;
float: left;
height: 270px;
margin-bottom: 15px;
}
#info_mesa_dado_antigo ul{
font-size: 12px;
}
#info_mesa_dado_antigo ul li{
list-style-type: none;
margin-bottom: 15px;
}
/* Informações dentro das contas antigas */
#info_mesa_dado_antigo ul li ol li{
list-style-type: decimal;
font-size: 10px;
margin-top: 3px;
margin-bottom: 2px;
color: #FF0000;
}
#info_mesa_rodape{

```



```

padding-top:10px;
text-align: center;
height: 12px;
float: none;
}
#loading_mesa{
position:absolute;
text-align: center;
margin: 0 auto;
margin-top: 2px;
width:840px;
}
#conta_mesa1, #conta_mesa2, #conta_mesa3, #conta_mesa4{
position:absolute;
/*z-index: 2;*/
border:0;
border-color: transparent;
margin-top: 35px;
margin-left: 20px;
}

#confirmar_fechamento{
position:absolute;
left:50%;
top:50%;
margin-left:-150px;
margin-top: -32.5px;
border: 1px solid #000000;
background-color:#FFFFFF;
color:black;
width: 300px;
height: 65px;
text-align: center;
visibility: hidden;
z-index: 3;
}

#confirmar_fechamento input{
border: 1px solid transparent;
border-right-color: #00807f;

```

```

border-bottom-color: #00807f;
background-color: transparent;
font-family: Verdana;
font-size: 10px;
}
#confirmar_fechamento input:hover{
color:red;
}
input.botaoOn {
border: 1px solid #1e599c;
background-color:#FFFFFF;
font-family: Verdana;
font-size: 11px;
color: #1e599c;
}
input.botaoOff {
border: 1px solid #777777;
color: #777777;
font-family: Verdana;
font-size: 11px;
background-color: #EFEFEF;
}

```

## ESTILOS.CSS

```

@IMPORT url("tabelaContaCozinha.css");
@IMPORT url("caixa.css");

body {
font-family: verdana, arial;
font-size: 11px;
margin: 0;
padding: 0;
text-align: center;
color:#1e599c;
/*background-color: #2369B8;*/
/*background-color: #72A7E4;*/
background-color: #23B3B8;
}

```

```
a {  
  color: #1e599c;  
  font-weight: bold;  
  text-decoration: none;  
}
```

```
a:link {  
  color: #1e599c;  
  font-weight: bold;  
  text-decoration: none;  
}
```

```
a:hover {  
  color: #1e599c;  
  text-decoration: underline;  
}
```

```
#principal {  
  width:840px;  
  text-align:left;  
  margin: 0 auto;  
  position: relative;  
  top: 0;  
  left: 0;  
  height: auto;  
  background-image: url("../img/principal_bg_ie.png");  
  background-repeat: repeat-y;
```

```
}  
body > #principal {  
  min-height: 640px;  
  height: auto;  
  background-image: url("../img/principal_bg.png");  
}
```

```
#cabecalho {  
  border-bottom: 1px solid #1e599c;  
  height: 166px;  
  margin-left: 10px;  
  margin-right: 10px;
```

```
background-image: url("../img/cabecalho_bg.png");
clear: both;
}
#rodape {
width: 840px;
text-align: left;
margin: 0 auto;
position: relative;
height: 38px;
background-image: url("../img/rodape_bg_ie.png");
clear: both;
}
```

```
body > #rodape {
background-image: url("../img/rodape_bg.png");
}
```

```
#rodape div {
margin-left: 10px;
margin-right: 10px;
border-top: 1px solid #283F08;
}
```

```
#rodape div p {
color: #959D9F;
text-align: center;
margin-top: 3px;
margin-bottom: 1px;
}
```

```
#sair_sistema {
margin: auto 0;
text-align:center;
position: relative;
bottom: -50px;
}
```

```
#conteudo {
margin: 20px;
}
```

```
#conteudo_caixa{
```

```

width: 654px;
height: 300px;
margin: 0 auto;
margin-top: 15px;
background-color: #FFFFFF;
}

```

```

#caixa_titulo{
text-align: center;
font-size: 14px;
font-weight: bold;

```

```

}
#caixa_titulo span{
border-bottom: 1px dashed;
}

```

```

#caixa_mesas {
margin: 0 auto;
margin-top: 15px;
width: 645px;
}

```

```

#caixa_mesas div{
float:left;
width: 155px;
height: 200px;
border-top: 1px solid;
border-left: 1px solid;
border-bottom: 1px solid;
margin-left: 2px;
}

```

```

#caixa_mesas div p{
margin-bottom: 15px;
text-align: center;
}

```

```

#caixa_mesas div p img{
cursor: pointer;
}

```

```

#caixa_mesas .mesaOn {

```

```
border: 2px solid green;  
}
```

```
#caixa_mesas .mesaOff {  
border: 2px solid red;  
}
```

```
#caixa_mesas .mesaPause {  
border: 2px solid #FF9900;  
}
```

```
#balao {  
width:242px;  
height:auto;  
position: absolute;  
visibility: hidden;  
top: 0;  
left: 0;  
z-index: 1;  
}
```

```
body > #balao {  
min-height:165px;  
height: auto;  
}
```

```
#balao_topo{  
background: url("../img/balao_01.png");  
height: 29px;  
width: 207px;  
float: left;  
}
```

```
#balao_fechar{  
background: url("../img/balao_02.png");  
height: 29px;  
width:35px;  
float:left;  
cursor: pointer;  
}
```

```
#balao_div {  
height: 29px;  
}
```

```

#balao_conteudo{
background: url("../img/balao_03.png") repeat-y;
height: 65px;
width: 242px;
text-align: center;
}
body > #balao_conteudo{
min-height: 180px;
height: auto;
}
#balao_fim {
background: url("../img/balao_04.png");
height: 56px;
text-align: center;
}
#balao_resposta{
color: red;
text-align: center;
}
#login {
width:420px;
border: 1px solid;
margin-left: 210px;
margin-top: 150px;
background-color: #FFFFFF;
}
#login .cabecalho {
text-align:center;
font-family: Verdana;
font-size: 12px;
color:#000000;
height:17px;
border-bottom: 1px solid #000000;
background-image: url("../img/login_cabecalho_bg.png");
}
#login_corpo {
margin-top: 10px;
}
#login_corpo div{

```

```

margin-bottom: 10px;
}
#login_corpo .label_login {
float: left;
margin-left: 105px;
}
#login_corpo span {
display: block;
width: 55px;
}
#login_corpo .input_login input {
border: 1px solid #eaebed;
padding-top: 3px;
padding-left: 4px;
font-family: Verdana;
font-size: 11px;
background-image: url("../img/fundo_texto.gif");
background-repeat: no-repeat;
}
#login_corpo .botao_login {
text-align: center;
}
#login_corpo .botao_login input{
border: 0;
font-family: Verdana;
font-size: 11px;
background-image: url("../img/login_botao_bg.png");
}
#erro_login {
margin-left: 140px;
border-top: 1px dotted #1e599c;
border-left: 1px dotted #1e599c;
border-right: 1px dotted #1e599c;
text-align: center;
color: red;
width: 150px;
height: 18px; padding-top: 8px;
}

```



```
#div_garcom input, select {
font-family: Verdana;
font-size: 10px;
}
#div_garcom p {
margin-top: 0px;
margin-bottom: 8px;
}
#balcao {
margin: 0 auto;
margin-top: 15px;
width: 645px;
}
#balcao div{
float:left;
width: 155px;
height: 160px;
border-top: 1px solid;
border-left: 1px solid;
border-bottom: 1px solid;
margin-left: 2px;
}
#balcao div p{
margin-bottom: 15px;
text-align: center;
}
#balcao div p img{
cursor: pointer;
}
#balcao .mesaOn {
border: 2px solid green;
}

#balcao .mesaOff {
border: 2px solid red;
}
#balcao .mesaPause {
border: 2px solid #FF9900;
}
```

```

/*
@media print {
div#barraVerticalEsquerda, div#barraMenuHorizontal {
    display: none;
}
div#conteudo { margin-left: auto; }
body#duasColunas { background: transparent; }
}
*/

```

## **PALMESTILOS.CSS**

```

body {
font-family: Verdana;
font-size: 12px;
}

#principal {
width: 240px;
}

#cabecalho {
border-bottom: 1px solid #1e599c;
height: 49px;
background-image: url("../img/cabecalho_pocket_bg.png");
background-repeat: no-repeat;
text-align: center;
display: block;
}

#loginPocket{
text-align: center;
border: 1px solid;
}

#loginPocket input, select{
font-family: Verdana;
font-size: 12px;
text-align: center;
}

.dataNascimento {
border: 0 none;
border-bottom: 1px solid black;
}

```

```

}

#produtos td, select{
font-size: 9px;
font-family: Verdana;
}

.botaoEnviar {
font-size: 9px;
font-family: Verdana;
}

#dados_mesa{
font-size: 10px;
font-family: Verdana;
}

#dados_mesa .mesaGarcom{
display: block;
float: left;
width: 150px;
}

#dados_mesa .nomeCliente{
text-align: right;
}

#tabelaPedidos{
width: auto;
}

#tabelaPedidos td{
font-size: 9px;
font-family: Verdana;
}

.botoes{
font-size: 9px;
font-family: Verdana;
}

.dadosConta {
text-align: center;
}

#tabelaContaCozinha table{
width: 100%;
}

```

```
#tabelaPedidos thead{
background-color: #23B3B8;
color: #FFFFFF;
font-weight: bold;
}
#tabelaPedidos tbody tr{
background-color: #F3F3F3;
}
#tabelaPedidos tfoot tr{
background-color: #FFFFFF;
text-align: right;
}
#tabelaPedidos .dadosFinal{
background-color: #F3F3F3;
text-align: center;
}
#sucessoPedido{
text-align: center;
font-size: 10px;
}
#sucessoPedido span {
display: block;
margin-top: 15px;
}
.padrao span {
text-align: center;
font-size: 10px;
display: block;
margin-top: 15px;
}
#liberarMesa{
font-size: 12px;
font-family: Verdana;
background-color: #FF7755;
}
#liberarMesa input {
font-size: 10px;
font-family: Verdana;
}
```

```
#liberarPedido{
font-size: 12px;
font-family: Verdana;
background-color: #FF7755;
}
#liberarPedido input {
font-size: 10px;
font-family: Verdana;
}
```

## **TABELACONTACOZINHA.CSS**

```
#tabelaContaCozinha table{
width: 100%;
}
#tabelaContaCozinha thead{
background-color: #23B3B8;
color: #FFFFFF;
font-weight: bold;
}
#tabelaContaCozinha tbody tr{
background-color: #F3F3F3;
}
#tabelaContaCozinha tfoot tr{
background-color: #FFFFFF;
text-align: right;
}
#tabelaContaCozinha tfoot td + td{
background-color: #F3F3F3;
text-align: left;
}
#botoes_listaPedidos{
margin-top:25px;
text-align: center;
}
```

## **AJAXREQUEST.JS**

```
/**
*
```

```

*
*/
function AjaxRequest(url) {

    // Private objects
    var xmlhttpRequest = createXMLHttpRequest();
    var requestURL = url;
    var queryString = null;
    var method = "GET";
    var preRequestFunction = null;
    var responseHandlerFunction = null;
    var textResponseHandlerFunction = null;
    var postRequestFunction = null;

    this.getXMLHttpRequest = function() {
        return xmlhttpRequest;
    }

    this.setPreRequestFunction = function(func) {
        preRequestFunction = func;
    }

    this.setResponseHandlerFunction = function(func) {
        responseHandlerFunction = func;
    }

    this.setTextResponseHandlerFunction = function(func) {
        textResponseHandlerFunction = func;
    }

    this.setPostRequestFunction = function(func) {
        postRequestFunction = func;
    }

    this.setMethod = function(met) {
        method = met.toUpperCase();
    }

    this.setUsePOSTMethod = function() {

```

```
        method = "POST";
    }

    this.setUseGETMethod = function() {
        method = "GET";
    }

    this.setQueryString = function(qs) {
        queryString = qs;
    }

    this.getQueryString = function() {
        return queryString;
    }

    this.addForm = function(form) {
        var qs = createQueryString(form.elements);
        appendQueryString(qs);
    }

    this.addFormById = function(formId) {
        var form = document.getElementById(formId);
        addForm(form);
    }

    this.addParameter = function(name, value) {
        var qs = name + "=" + value;
        appendQueryString(qs);
    }

    this.addFormFieldByName = function() {
        var elementName = null;
        var elements = null;
        var qs = null;
        for (var i = 0; i < arguments.length; i++) {
            elementName = arguments[i];
            elements = document.getElementsByName(elementName);
            qs = createQueryString(elements);
            appendQueryString(qs);
        }
    }
}
```

```

    }
}

this.addFormFieldById = function() {
    var element = null;
    var elements = new Array(arguments.length);
    for (var i = 0; i < arguments.length; i++) {
        element = document.getElementById(arguments[i]);
        if (element != null) {
            elements[i] = element;
        }
    }
    qs = createQueryString(elements);
    appendQueryString(qs);
}

this.send = function() {
    if (preRequestFunction) {
        preRequestFunction(this);
    }

    var thisObj = this;
    xmlhttpRequest.onreadystatechange = function() {
        handleResponse(thisObj);
    };

    if (method.toUpperCase() == "GET") {
        if (queryString != null && queryString.length > 0) {
            if (requestURL.indexOf("?") != -1) {
                requestURL = requestURL + "&" + queryString;
            } else {
                requestURL = requestURL + "?" + queryString;
            }
        }
        xmlhttpRequest.open(method, encodeURI(requestURL), true);
        xmlhttpRequest.send(null);
    } else {
        xmlhttpRequest.open(method, encodeURI(requestURL), true);
    }
}

```



```

        xmlhttpRequest.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

        xmlhttpRequest.send(queryString);
    }
}

appendQueryString = function(str) {
    if (queryString != null && queryString.length > 0) {
        queryString = queryString + "&" + str;
    } else {
        queryString = str;
    }
}

createQueryString = function(elements) {
    var tempQueryString = "";
    var temp = "";
    var name = null;
    var formElement = null;
    for (var i = 0; i < elements.length; i++) {
        formElement = elements[i];
        name = formElement.getAttribute("name");

        if (formElement.tagName.toLowerCase() == "input") {
            if ((formElement.type.toLowerCase() == "radio"
                || formElement.type.toLowerCase() == "checkbox")
                && (formElement.checked)) {
                temp = name + "=" + formElement.value;
            } else if (formElement.type.toLowerCase() != "submit"
                && formElement.type.toLowerCase() !=
"button"
                && formElement.type.toLowerCase() !=
"reset") {
                temp = name + "=" + formElement.value;
            }
        } else if (formElement.tagName.toLowerCase() == "textarea") {
            temp = name + "=" + formElement.value;
        } else if (formElement.tagName.toLowerCase() == "select") {
            var options = formElement.options;

```

```

        var optionsTempQS = "";
        for (var j = 0; j < options.length; j++) {
            if (options[j].selected) {
                if (optionsTempQS.length > 0) {
                    optionsTempQS = optionsTempQS + "&" +
name + "=" + options[j].value;
                } else {
                    optionsTempQS = name + "=" +
options[j].value;
                }
            }
        }
        temp = optionsTempQS;
    }

    if (temp != null && temp.length > 0) {
        if (tempQueryString != null && tempQueryString.length > 0) {
            tempQueryString = tempQueryString + "&" + temp;
        } else {
            tempQueryString = temp;
        }
    }
    return tempQueryString;
}

```

```

handleResponse = function(ajaxRequest) {
    var req = ajaxRequest.getXMLHttpRequest();
    // readyState 4 means 'complete'
    if (req.readyState != 4) {
        return false;
    }
    if (req.status == 200) {
        if (responseHandlerFunction) {
            responseHandlerFunction(req.responseXML);
        }
        if (textResponseHandlerFunction) {
            textResponseHandlerFunction(req.responseText);
        }
    }
}

```

```

        if (postRequestFunction) {
            postRequestFunction(ajaxRequest);
        }
        return true;
    } else {
        return false;
    }
}

}

/**
 * Creates a new XMLHttpRequest object in any browser that supports it.
 */
function createXMLHttpRequest() {
    var request = false;

    if (window.XMLHttpRequest) {
        // Create XMLHttpRequest object in non-Microsoft browsers
        request = new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        // Create XMLHttpRequest via MS ActiveX
        try {
            // Try to create XMLHttpRequest in later versions
            // of Internet Explorer
            request = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (ex1) {
            // Failed to create required ActiveXObject
            try {
                // Try version supported by older versions
                // of Internet Explorer
                request = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (ex2) {
                // Unable to create an XMLHttpRequest with ActiveX
            }
        }
    }

    return request;
}

```

## FUNCOES.JS

```

function logar(){
    var form = $("form_login");
    $("resposta_lendo").style.visibility= "visible";
    ajaxReq = new AjaxRequest(form.action);
    ajaxReq.addForm(form);
    ajaxReq.setTextResponseHandlerFunction(receberResultado);
    ajaxReq.send();
}

function correcao(dado){
    if(dado != null && dado != ""){
        if(dado.indexOf("mesaOn") == -1 && dado.indexOf("mesaOff") == -1 &&
dado.indexOf("mesaPause") == -1)
            return true;
    }
    return false;
}

function receberResultado(resposta){
    $("resposta_lendo").style.visibility= "hidden";
    if(resposta == "false"){
        erro();
    }else{
        if(resposta == "1"){
            window.location="menu_balcao.php";
        }else if(resposta == "2"){
            window.location="menu_caixa.php";
        }else{
            alert(resposta);
        }
    }
}

function esconderBalao(){
    var balaoDiv = $("balao");
    var balao_resposta = $("balao_resposta");
    var balao_conteudo = $("balao_conteudo");
    balao_conteudo.innerHTML = "";
    Element.setStyle(balaoDiv, {visibility: "hidden"});
}

```

```

Element.setStyle(balao_resposta, {visibility: "hidden"});
}
function acessoNegado(mensagem){
    $("erro_login").style.visibility= "visible";
    $("erro_login").innerHTML = mensagem;
    Effect.BlindDown('erro_login');
    var desaparecer = function(){
        return new Effect.Fade('erro_login');
    };
    timeoutStatusId = window.setTimeout(desaparecer, 5000);
}
function getPosition(theElement) {
    var positionX = 0;
    var positionY = 0;

    while (theElement != null) {
        positionX += theElement.offsetLeft;
        positionY += theElement.offsetTop;
        theElement = theElement.offsetParent;
    }
    return {x: positionX, y: positionY};
}

function url_encode(str) {
    var hex_chars = "0123456789ABCDEF";
    var noEncode = /^[a-zA-Z0-9_\-\.\$]/;
    var n, strCode, hex1, hex2, strEncode = "";

    for(n = 0; n < str.length; n++) {
        if (noEncode.test(str.charAt(n))) {
            strEncode += str.charAt(n);
        } else {
            strCode = str.charCodeAt(n);
            hex1 = hex_chars.charAt(Math.floor(strCode / 16));
            hex2 = hex_chars.charAt(strCode % 16);
            strEncode += "%" + (hex1 + hex2);
        }
    }
}

```

```

return strEncode; }
// url_decode version 1.0
function url_decode(str) {
    var n, strCode, strDecode = "";
    for (n = 0; n < str.length; n++) {
        if (str.charAt(n) == "%") {
            strCode = str.charAt(n + 1) + str.charAt(n + 2);
            strDecode += String.fromCharCode(parseInt(strCode, 16));
            n += 2;
        } else {
            strDecode += str.charAt(n);
        }
    }
    return strDecode;
}
function fecharDetalheMesa(){
Effect.BlindUp('info_mesa');
$("info_mesa").innerHTML = "";
}

```

## PROTOTYPE.JS

```

/* Prototype JavaScript framework, version 1.5.0_rc0
 * (c) 2005 Sam Stephenson <sam@conio.net>
 *
 * Prototype is freely distributable under the terms of an MIT-style license.
 * For details, see the Prototype web site: http://prototype.conio.net/
 *
 */

```

```

var Prototype = {
  Version: '1.5.0_rc0',
  ScriptFragment: '(?:<script.*?>)((\\n|\\r|.)*?)(?:</script>)',

  emptyFunction: function() {},
  K: function(x) {return x}
}

```

```

var Class = {
  create: function() {
    return function() {

```

```

        this.initialize.apply(this, arguments);
    }
}
}

var Abstract = new Object();

Object.extend = function(destination, source) {
    for (var property in source) {
        destination[property] = source[property];
    }
    return destination;
}

Object.inspect = function(object) {
    try {
        if (object == undefined) return 'undefined';
        if (object == null) return 'null';
        return object.inspect ? object.inspect() : object.toString();
    } catch (e) {
        if (e instanceof RangeError) return '...';
        throw e;
    }
}

Function.prototype.bind = function() {
    var __method = this, args = $A(arguments), object = args.shift();
    return function() {
        return __method.apply(object, args.concat($A(arguments)));
    }
}

Function.prototype.bindAsEventListener = function(object) {
    var __method = this;
    return function(event) {
        return __method.call(object, event || window.event);
    }
}

```

```

Object.extend(Number.prototype, {
  toColorPart: function() {
    var digits = this.toString(16);
    if (this < 16) return '0' + digits;
    return digits;
  },

  succ: function() {
    return this + 1;
  },

  times: function(iterator) {
    $R(0, this, true).each(iterator);
    return this;
  }
});

var Try = {
  these: function() {
    var returnValue;

    for (var i = 0; i < arguments.length; i++) {
      var lambda = arguments[i];
      try {
        returnValue = lambda();
        break;
      } catch (e) {}
    }

    return returnValue;
  }
}

/*-----*/

var PeriodicalExecuter = Class.create();
PeriodicalExecuter.prototype = {
  initialize: function(callback, frequency) {
    this.callback = callback;
  }
}

```



```

    this.frequency = frequency;
    this.currentlyExecuting = false;

    this.registerCallback();
  },

  registerCallback: function() {
    setInterval(this.onTimerEvent.bind(this), this.frequency * 1000);
  },

  onTimerEvent: function() {
    if (!this.currentlyExecuting) {
      try {
        this.currentlyExecuting = true;
        this.callback();
      } finally {
        this.currentlyExecuting = false;
      }
    }
  }
}

Object.extend(String.prototype, {
  gsub: function(pattern, replacement) {
    var result = "", source = this, match;
    replacement = arguments.callee.prepareReplacement(replacement);

    while (source.length > 0) {
      if (match = source.match(pattern)) {
        result += source.slice(0, match.index);
        result += (replacement(match) || "").toString();
        source = source.slice(match.index + match[0].length);
      } else {
        result += source, source = "";
      }
    }
    return result;
  },

  sub: function(pattern, replacement, count) {

```

```

replacement = this.gsub.prepareReplacement(replacement);
count = count === undefined ? 1 : count;

return this.gsub(pattern, function(match) {
  if (--count < 0) return match[0];
  return replacement(match);
});
},

scan: function(pattern, iterator) {
  this.gsub(pattern, iterator);
  return this;
},

truncate: function(length, truncation) {
  length = length || 30;
  truncation = truncation === undefined ? '...' : truncation;
  return this.length > length ?
    this.slice(0, length - truncation.length) + truncation : this;
},

strip: function() {
  return this.replace(/^\s+/, "").replace(/\s+$/, "");
},

stripTags: function() {
  return this.replace(/<V?[^>]+>/gi, "");
},

stripScripts: function() {
  return this.replace(new RegExp(Prototype.ScriptFragment, 'img'), "");
},

extractScripts: function() {
  var matchAll = new RegExp(Prototype.ScriptFragment, 'img');
  var matchOne = new RegExp(Prototype.ScriptFragment, 'im');
  return (this.match(matchAll) || []).map(function(scriptTag) {
    return (scriptTag.match(matchOne) || ['', ''])[1];
  });
}

```

```

},

evalScripts: function() {
    return this.extractScripts().map(function(script) { return eval(script) });
},

escapeHTML: function() {
    var div = document.createElement('div');
    var text = document.createTextNode(this);
    div.appendChild(text);
    return div.innerHTML;
},

unescapeHTML: function() {
    var div = document.createElement('div');
    div.innerHTML = this.stripTags();
    return div.childNodes[0] ? div.childNodes[0].nodeValue : "";
},

toQueryParams: function() {
    var pairs = this.match(/^(?:.*)"$/)[1].split('&');
    return pairs.inject({}, function(params, pairString) {
        var pair = pairString.split('=');
        params[pair[0]] = pair[1];
        return params;
    });
},

toArray: function() {
    return this.split("");
},

camelize: function() {
    var oStringList = this.split('-');
    if (oStringList.length == 1) return oStringList[0];

    var camelizedString = this.indexOf('-') == 0
        ? oStringList[0].charAt(0).toUpperCase() + oStringList[0].substring(1)
        : oStringList[0];

```

```

    for (var i = 1, len = oStringList.length; i < len; i++) {
        var s = oStringList[i];
        camelizedString += s.charAt(0).toUpperCase() + s.substring(1);
    }

    return camelizedString;
},

inspect: function() {
    return "" + this.replace(/\\g, '\\\\').replace(/"/g, "\\") + "";
}
});

String.prototype.gsub.prepareReplacement = function(replacement) {
    if (typeof replacement == 'function') return replacement;
    var template = new Template(replacement);
    return function(match) { return template.evaluate(match) };
}

String.prototype.parseQuery = String.prototype.toQueryParams;

var Template = Class.create();
Template.Pattern = /^(^|.|\\r\\n)(#\\{(.*)\\})/;
Template.prototype = {
    initialize: function(template, pattern) {
        this.template = template.toString();
        this.pattern = pattern || Template.Pattern;
    },

    evaluate: function(object) {
        return this.template.gsub(this.pattern, function(match) {
            var before = match[1];
            if (before == '\\') return match[2];
            return before + (object[match[3]] || "").toString();
        });
    }
}

```

```

var $break = new Object();
var $continue = new Object();

```

```

var Enumerable = {
  each: function(iterator) {
    var index = 0;
    try {
      this._each(function(value) {
        try {
          iterator(value, index++);
        } catch (e) {
          if (e != $continue) throw e;
        }
      });
    } catch (e) {
      if (e != $break) throw e;
    }
  },

```

```

  all: function(iterator) {
    var result = true;
    this.each(function(value, index) {
      result = result && !(iterator || Prototype.K)(value, index);
      if (!result) throw $break;
    });
    return result;
  },

```

```

  any: function(iterator) {
    var result = true;
    this.each(function(value, index) {
      if (result = !(iterator || Prototype.K)(value, index))
        throw $break;
    });
    return result;
  },

```

```

  collect: function(iterator) {
    var results = [];

```

```

    this.each(function(value, index) {
        results.push(iterator(value, index));
    });
    return results;
},

```

```

detect: function (iterator) {
    var result;
    this.each(function(value, index) {
        if (iterator(value, index)) {
            result = value;
            throw $break;
        }
    });
    return result;
},

```

```

findAll: function(iterator) {
    var results = [];
    this.each(function(value, index) {
        if (iterator(value, index))
            results.push(value);
    });
    return results;
},

```

```

grep: function(pattern, iterator) {
    var results = [];
    this.each(function(value, index) {
        var stringValue = value.toString();
        if (stringValue.match(pattern))
            results.push((iterator || Prototype.K)(value, index));
    })
    return results;
},

```

```

include: function(object) {
    var found = false;
    this.each(function(value) {

```

```

    if (value == object) {
      found = true;
      throw $break;
    }
  });
  return found;
},

```

```

inject: function(memo, iterator) {
  this.each(function(value, index) {
    memo = iterator(memo, value, index);
  });
  return memo;
},

```

```

invoke: function(method) {
  var args = $A(arguments).slice(1);
  return this.collect(function(value) {
    return value[method].apply(value, args);
  });
},

```

```

max: function(iterator) {
  var result;
  this.each(function(value, index) {
    value = (iterator || Prototype.K)(value, index);
    if (result == undefined || value >= result)
      result = value;
  });
  return result;
},

```

```

min: function(iterator) {
  var result;
  this.each(function(value, index) {
    value = (iterator || Prototype.K)(value, index);
    if (result == undefined || value < result)
      result = value;
  });
},

```

```
    return result;
  },
```

```
partition: function(iterator) {
  var trues = [], falses = [];
  this.each(function(value, index) {
    ((iterator || Prototype.K)(value, index) ?
      trues : falses).push(value);
  });
  return [trues, falses];
},
```

```
pluck: function(property) {
  var results = [];
  this.each(function(value, index) {
    results.push(value[property]);
  });
  return results;
},
```

```
reject: function(iterator) {
  var results = [];
  this.each(function(value, index) {
    if (!iterator(value, index))
      results.push(value);
  });
  return results;
},
```

```
sortBy: function(iterator) {
  return this.collect(function(value, index) {
    return {value: value, criteria: iterator(value, index)};
  }).sort(function(left, right) {
    var a = left.criteria, b = right.criteria;
    return a < b ? -1 : a > b ? 1 : 0;
  }).pluck('value');
},
```

```
toArray: function() {
```



```

    return this.collect(Prototype.K);
  },

  zip: function() {
    var iterator = Prototype.K, args = $A(arguments);
    if (typeof args.last() == 'function')
      iterator = args.pop();

    var collections = [this].concat(args).map($A);
    return this.map(function(value, index) {
      return iterator(collections.pluck(index));
    });
  },

  inspect: function() {
    return '#<Enumerable:' + this.toArray().inspect() + '>';
  }
}

Object.extend(Enumerable, {
  map:    Enumerable.collect,
  find:   Enumerable.detect,
  select: Enumerable.findAll,
  member: Enumerable.include,
  entries: Enumerable.toArray
});

var $A = Array.from = function(iterable) {
  if (!iterable) return [];
  if (iterable.toArray) {
    return iterable.toArray();
  } else {
    var results = [];
    for (var i = 0; i < iterable.length; i++)
      results.push(iterable[i]);
    return results;
  }
}

Object.extend(Array.prototype, Enumerable);

```

```

if (!Array.prototype._reverse)
  Array.prototype._reverse = Array.prototype.reverse;

Object.extend(Array.prototype, {
  _each: function(iterator) {
    for (var i = 0; i < this.length; i++)
      iterator(this[i]);
  },

  clear: function() {
    this.length = 0;
    return this;
  },

  first: function() {
    return this[0];
  },

  last: function() {
    return this[this.length - 1];
  },

  compact: function() {
    return this.select(function(value) {
      return value != undefined || value != null;
    });
  },

  flatten: function() {
    return this.inject([], function(array, value) {
      return array.concat(value && value.constructor == Array ?
        value.flatten() : [value]);
    });
  },

  without: function() {
    var values = $A(arguments);
    return this.select(function(value) {

```

```

        return !values.include(value);
    });
},

indexOf: function(object) {
    for (var i = 0; i < this.length; i++)
        if (this[i] == object) return i;
    return -1;
},

reverse: function(inline) {
    return (inline !== false ? this : this.toArray())._reverse();
},

inspect: function() {
    return '[' + this.map(Object.inspect).join(', ') + ']';
}
});

var Hash = {
    _each: function(iterator) {
        for (var key in this) {
            var value = this[key];
            if (typeof value == 'function') continue;

            var pair = [key, value];
            pair.key = key;
            pair.value = value;
            iterator(pair);
        }
    },

    keys: function() {
        return this.pluck('key');
    },

    values: function() {
        return this.pluck('value');
    },

```

```

merge: function(hash) {
  return $H(hash).inject($H(this), function(mergedHash, pair) {
    mergedHash[pair.key] = pair.value;
    return mergedHash;
  });
},

```

```

toQueryString: function() {
  return this.map(function(pair) {
    return pair.map(encodeURIComponent).join('=');
  }).join('&');
},

```

```

inspect: function() {
  return '#<Hash:{' + this.map(function(pair) {
    return pair.map(Object.inspect).join(': ');
  }).join(', ' + '>';
}
}

```

```

function $H(object) {
  var hash = Object.extend({}, object || {});
  Object.extend(hash, Enumerable);
  Object.extend(hash, Hash);
  return hash;
}

ObjectRange = Class.create();
Object.extend(ObjectRange.prototype, Enumerable);
Object.extend(ObjectRange.prototype, {
  initialize: function(start, end, exclusive) {
    this.start = start;
    this.end = end;
    this.exclusive = exclusive;
  },

```

```

  _each: function(iterator) {
    var value = this.start;
    do {
      iterator(value);

```

```

        value = value.succ();
    } while (this.include(value));
},

include: function(value) {
    if (value < this.start)
        return false;
    if (this.exclusive)
        return value < this.end;
    return value <= this.end;
}
});

var $R = function(start, end, exclusive) {
    return new ObjectRange(start, end, exclusive);
}

var Ajax = {
    getTransport: function() {
        return Try.these(
            function() {return new XMLHttpRequest();},
            function() {return new ActiveXObject('Msxml2.XMLHTTP');},
            function() {return new ActiveXObject('Microsoft.XMLHTTP')}
        ) || false;
    },

    activeRequestCount: 0
}

Ajax.Responders = {
    responders: [],

    _each: function(iterator) {
        this.responders._each(iterator);
    },

    register: function(responderToAdd) {
        if (!this.include(responderToAdd))
            this.responders.push(responderToAdd);
    }
};

```

```

},

unregister: function(responderToRemove) {
    this.responders = this.responders.without(responderToRemove);
},

dispatch: function(callback, request, transport, json) {
    this.each(function(responder) {
        if (responder[callback] && typeof responder[callback] == 'function') {
            try {
                responder[callback].apply(responder, [request, transport, json]);
            } catch (e) {}
        }
    });
}

};

Object.extend(Ajax.Responders, Enumerable);

Ajax.Responders.register({
    onCreate: function() {
        Ajax.activeRequestCount++;
    },

    onComplete: function() {
        Ajax.activeRequestCount--;
    }
});

Ajax.Base = function() {};
Ajax.Base.prototype = {
    setOptions: function(options) {
        this.options = {
            method: 'post',
            asynchronous: true,
            contentType: 'application/x-www-form-urlencoded',
            parameters: ''
        }
        Object.extend(this.options, options || {});
    }
};

```

```

},

responselsSuccess: function() {
    return this.transport.status == undefined
        || this.transport.status == 0
        || (this.transport.status >= 200 && this.transport.status < 300);
},

responselsFailure: function() {
    return !this.responselsSuccess();
}
}

Ajax.Request = Class.create();
Ajax.Request.Events =
    ['Uninitialized', 'Loading', 'Loaded', 'Interactive', 'Complete'];

Ajax.Request.prototype = Object.extend(new Ajax.Base(), {
    initialize: function(url, options) {
        this.transport = Ajax.getTransport();
        this.setOptions(options);
        this.request(url);
    },

    request: function(url) {
        var parameters = this.options.parameters || '';
        if (parameters.length > 0) parameters += '&_=';

        try {
            this.url = url;
            if (this.options.method == 'get' && parameters.length > 0)
                this.url += (this.url.match(/\?/) ? '&' : '?') + parameters;

            Ajax.Responders.dispatch('onCreate', this, this.transport);

            this.transport.open(this.options.method, this.url,
                this.options.asynchronous);

            if (this.options.asynchronous) {

```

```

        this.transport.onreadystatechange = this.onStateChange.bind(this);
        setTimeout((function() {this.respondToReadyState(1)}).bind(this), 10);
    }

    this.setRequestHeader();

    var body = this.options.postBody ? this.options.postBody : parameters;
    this.transport.send(this.options.method == 'post' ? body : null);

} catch (e) {
    this.dispatchEvent(e);
}
},

setRequestHeaders: function() {
    var requestHeaders =
        ['X-Requested-With', 'XMLHttpRequest',
        'X-Prototype-Version', Prototype.Version,
        'Accept', 'text/javascript, text/html, application/xml, text/xml, */*'];

    if (this.options.method == 'post') {
        requestHeaders.push('Content-type', this.options.contentType);

        /* Force "Connection: close" for Mozilla browsers to work around
        * a bug where XMLHttpRequest sends an incorrect Content-length
        * header. See Mozilla Bugzilla #246651.
        */
        if (this.transport.overrideMimeType)
            requestHeaders.push('Connection', 'close');
    }

    if (this.options.requestHeaders)
        requestHeaders.push.apply(requestHeaders, this.options.requestHeaders);

    for (var i = 0; i < requestHeaders.length; i += 2)
        this.transport.setRequestHeader(requestHeaders[i], requestHeaders[i+1]);
},

onStateChange: function() {

```



```

    var readyState = this.transport.readyState;
    if (readyState != 1)
        this.respondToReadyState(this.transport.readyState);
},

header: function(name) {
    try {
        return this.transport.getResponseHeader(name);
    } catch (e) {}
},

evalJSON: function() {
    try {
        return eval('(' + this.header('X-JSON') + ')');
    } catch (e) {}
},

evalResponse: function() {
    try {
        return eval(this.transport.responseText);
    } catch (e) {
        this.dispatchException(e);
    }
},

respondToReadyState: function(readyState) {
    var event = Ajax.Request.Events[readyState];
    var transport = this.transport, json = this.evalJSON();

    if (event == 'Complete') {
        try {
            (this.options['on' + this.transport.status]
             || this.options['on' + (this.respondsSuccess() ? 'Success' : 'Failure')]
             || Prototype.emptyFunction)(transport, json);
        } catch (e) {
            this.dispatchException(e);
        }

        if ((this.header('Content-type') || '').match(/^text\/javascript/i))

```

```

        this.evalResponse();
    }

    try {
        (this.options['on' + event] || Prototype.emptyFunction)(transport, json);
        Ajax.Responders.dispatch('on' + event, this, transport, json);
    } catch (e) {
        this.dispatchException(e);
    }

    /* Avoid memory leak in MSIE: clean up the oncomplete event handler */
    if (event == 'Complete')
        this.transport.onreadystatechange = Prototype.emptyFunction;
},

dispatchException: function(exception) {
    (this.options.onException || Prototype.emptyFunction)(this, exception);
    Ajax.Responders.dispatch('onException', this, exception);
}
});

Ajax.Updater = Class.create();

Object.extend(Object.extend(Ajax.Updater.prototype, Ajax.Request.prototype), {
    initialize: function(container, url, options) {
        this.containers = {
            success: container.success ? $(container.success) : $(container),
            failure: container.failure ? $(container.failure) :
                (container.success ? null : $(container))
        }

        this.transport = Ajax.getTransport();
        this.setOptions(options);

        var onComplete = this.options.onComplete || Prototype.emptyFunction;
        this.options.onComplete = (function(transport, object) {
            this.updateContent();
            onComplete(transport, object);
        }).bind(this);
    }
});

```

```

    this.request(url);
},

updateContent: function() {
    var receiver = this.responsesSuccess() ?
        this.containers.success : this.containers.failure;
    var response = this.transport.responseText;

    if (!this.options.evalScripts)
        response = response.stripScripts();

    if (receiver) {
        if (this.options.insertion) {
            new this.options.insertion(receiver, response);
        } else {
            Element.update(receiver, response);
        }
    }

    if (this.responsesSuccess()) {
        if (this.onComplete)
            setTimeout(this.onComplete.bind(this), 10);
    }
}

});

Ajax.PeriodicalUpdater = Class.create();
Ajax.PeriodicalUpdater.prototype = Object.extend(new Ajax.Base(), {
    initialize: function(container, url, options) {
        this.setOptions(options);
        this.onComplete = this.options.onComplete;

        this.frequency = (this.options.frequency || 2);
        this.decay = (this.options.decay || 1);

        this.updater = {};
        this.container = container;
        this.url = url;

```

```

        this.start();
    },

    start: function() {
        this.options.onComplete = this.updateComplete.bind(this);
        this.onTimerEvent();
    },

    stop: function() {
        this.updater.onComplete = undefined;
        clearTimeout(this.timer);
        (this.onComplete || Prototype.emptyFunction).apply(this, arguments);
    },

    updateComplete: function(request) {
        if (this.options.decay) {
            this.decay = (request.responseText == this.lastText ?
                this.decay * this.options.decay : 1);

            this.lastText = request.responseText;
        }
        this.timer = setTimeout(this.onTimerEvent.bind(this),
            this.decay * this.frequency * 1000);
    },

    onTimerEvent: function() {
        this.updater = new Ajax.Updater(this.container, this.url, this.options);
    }
});

function $() {
    var results = [], element;
    for (var i = 0; i < arguments.length; i++) {
        element = arguments[i];
        if (typeof element == 'string')
            element = document.getElementById(element);
        results.push(Element.extend(element));
    }
    return results.length < 2 ? results[0] : results;
}

```

```

}

document.getElementsByClassName = function(className, parentElement) {
    var children = ($(parentElement) || document.body).getElementsByTagName('*');
    return $A(children).inject([], function(elements, child) {
        if (child.className.match(new RegExp("(^|\\s)" + className + "(\\s|$)")))
            elements.push(Element.extend(child));
        return elements;
    });
}

/*-----*/

if (!window.Element)
    var Element = new Object();

Element.extend = function(element) {
    if (!element) return;
    if (_nativeExtensions) return element;

    if (!element._extended && element.tagName && element != window) {
        var methods = Element.Methods, cache = Element.extend.cache;
        for (property in methods) {
            var value = methods[property];
            if (typeof value == 'function')
                element[property] = cache.findOrStore(value);
        }
    }

    element._extended = true;
    return element;
}

Element.extend.cache = {
    findOrStore: function(value) {
        return this[value] = this[value] || function() {
            return value.apply(null, [this].concat($A(arguments)));
        }
    }
}

```

```
}
```

```
Element.Methods = {
```

```
  visible: function(element) {
    return $(element).style.display != 'none';
  },
```

```
  toggle: function() {
    for (var i = 0; i < arguments.length; i++) {
      var element = $(arguments[i]);
      Element[Element.visible(element) ? 'hide' : 'show'](element);
    }
  },
```

```
  hide: function() {
    for (var i = 0; i < arguments.length; i++) {
      var element = $(arguments[i]);
      element.style.display = 'none';
    }
  },
```

```
  show: function() {
    for (var i = 0; i < arguments.length; i++) {
      var element = $(arguments[i]);
      element.style.display = "";
    }
  },
```

```
  remove: function(element) {
    element = $(element);
    element.parentNode.removeChild(element);
  },
```

```
  update: function(element, html) {
    $(element).innerHTML = html.stripScripts();
    setTimeout(function() {html.evalScripts()}, 10);
  },
```

```
  replace: function(element, html) {
```

```

    element = $(element);
    if (element.outerHTML) {
        element.outerHTML = html.stripScripts();
    } else {
        var range = element.ownerDocument.createRange();
        range.selectNodeContents(element);
        element.parentNode.replaceChild(
            range.createContextualFragment(html.stripScripts()), element);
    }
    setTimeout(function() {html.evalScripts()}, 10);
},

```

```

getHeight: function(element) {
    element = $(element);
    return element.offsetHeight;
},

```

```

classNames: function(element) {
    return new Element.ClassNames(element);
},

```

```

hasClassName: function(element, className) {
    if (!(element = $(element))) return;
    return Element.classNames(element).include(className);
},

```

```

addClassName: function(element, className) {
    if (!(element = $(element))) return;
    return Element.classNames(element).add(className);
},

```

```

removeClassName: function(element, className) {
    if (!(element = $(element))) return;
    return Element.classNames(element).remove(className);
},

```

// removes whitespace-only text node children

```

cleanWhitespace: function(element) {
    element = $(element);

```

```

for (var i = 0; i < element.childNodes.length; i++) {
  var node = element.childNodes[i];
  if (node.nodeType == 3 && /\S/.test(node.nodeValue))
    Element.remove(node);
}
},

empty: function(element) {
  return $(element).innerHTML.match(/^\s*$/);
},

childOf: function(element, ancestor) {
  element = $(element), ancestor = $(ancestor);
  while (element = element.parentNode)
    if (element == ancestor) return true;
  return false;
},

scrollTo: function(element) {
  element = $(element);
  var x = element.x ? element.x : element.offsetLeft,
      y = element.y ? element.y : element.offsetTop;
  window.scrollTo(x, y);
},

getStyle: function(element, style) {
  element = $(element);
  var value = element.style[style.camelize()];
  if (!value) {
    if (document.defaultView && document.defaultView.getComputedStyle) {
      var css = document.defaultView.getComputedStyle(element, null);
      value = css ? css.getPropertyValue(style) : null;
    } else if (element.currentStyle) {
      value = element.currentStyle[style.camelize()];
    }
  }
}

if (window.opera && ['left', 'top', 'right', 'bottom'].include(style))
  if (Element.getStyle(element, 'position') == 'static') value = 'auto';

```



```

    return value == 'auto' ? null : value;
},

setStyle: function(element, style) {
    element = $(element);
    for (var name in style)
        element.style[name.camelize()] = style[name];
},

getDimensions: function(element) {
    element = $(element);
    if (Element.getStyle(element, 'display') != 'none')
        return {width: element.offsetWidth, height: element.offsetHeight};

    // All *Width and *Height properties give 0 on elements with display none,
    // so enable the element temporarily
    var els = element.style;
    var originalVisibility = els.visibility;
    var originalPosition = els.position;
    els.visibility = 'hidden';
    els.position = 'absolute';
    els.display = "";
    var originalWidth = element.clientWidth;
    var originalHeight = element.clientHeight;
    els.display = 'none';
    els.position = originalPosition;
    els.visibility = originalVisibility;
    return {width: originalWidth, height: originalHeight};
},

makePositioned: function(element) {
    element = $(element);
    var pos = Element.getStyle(element, 'position');
    if (pos == 'static' || !pos) {
        element._madePositioned = true;
        element.style.position = 'relative';
        // Opera returns the offset relative to the positioning context, when an
        // element is position relative but top and left have not been defined
    }
}

```

```

    if (window.opera) {
        element.style.top = 0;
        element.style.left = 0;
    }
}
},

```

```

undoPositioned: function(element) {
    element = $(element);
    if (element._madePositioned) {
        element._madePositioned = undefined;
        element.style.position =
            element.style.top =
            element.style.left =
            element.style.bottom =
            element.style.right = "";
    }
},

```

```

makeClipping: function(element) {
    element = $(element);
    if (element._overflow) return;
    element._overflow = element.style.overflow;
    if ((Element.getStyle(element, 'overflow') || 'visible') != 'hidden')
        element.style.overflow = 'hidden';
},

```

```

undoClipping: function(element) {
    element = $(element);
    if (element._overflow) return;
    element.style.overflow = element._overflow;
    element._overflow = undefined;
}
}

```

```
Object.extend(Element, Element.Methods);
```

```
var _nativeExtensions = false;
```

```

if(!HTMLElement && /Konqueror|Safari|KHTML/.test(navigator.userAgent)) {
  var HTMLElement = {}
  HTMLElement.prototype = document.createElement('div').__proto__;
}

```

```

Element.addMethods = function(methods) {
  Object.extend(Element.Methods, methods || {});

```

```

if(typeof HTMLElement != 'undefined') {
  var methods = Element.Methods, cache = Element.extend.cache;
  for (property in methods) {
    var value = methods[property];
    if (typeof value == 'function')
      HTMLElement.prototype[property] = cache.findOrStore(value);
  }
  _nativeExtensions = true;
}
}

```

```

Element.addMethods();

```

```

var Toggle = new Object();
Toggle.display = Element.toggle;

```

```

/*-----*/

```

```

Abstract.Insertion = function(adjacency) {
  this.adjacency = adjacency;
}

```

```

Abstract.Insertion.prototype = {
  initialize: function(element, content) {
    this.element = $(element);
    this.content = content.stripScripts();

    if (this.adjacency && this.element.insertAdjacentHTML) {
      try {
        this.element.insertAdjacentHTML(this.adjacency, this.content);
      } catch (e) {

```

```

        var tagName = this.element.tagName.toLowerCase();
        if (tagName == 'tbody' || tagName == 'tr') {
            this.insertContent(this.contentFromAnonymousTable());
        } else {
            throw e;
        }
    }
} else {
    this.range = this.element.ownerDocument.createRange();
    if (this.initializeRange) this.initializeRange();
    this.insertContent([this.range.createContextualFragment(this.content)]);
}

setTimeout(function() {content.evalScripts()}, 10);
},

contentFromAnonymousTable: function() {
    var div = document.createElement('div');
    div.innerHTML = '<table><tbody>' + this.content + '</tbody></table>';
    return $A(div.childNodes[0].childNodes[0].childNodes);
}
}

var Insertion = new Object();

Insertion.Before = Class.create();
Insertion.Before.prototype = Object.extend(new Abstract.Insertion('beforeBegin'), {
    initializeRange: function() {
        this.range.setStartBefore(this.element);
    },

    insertContent: function(fragments) {
        fragments.each((function(fragment) {
            this.element.parentNode.insertBefore(fragment, this.element);
        }).bind(this));
    }
});

Insertion.Top = Class.create();

```

```

Insertion.Top.prototype = Object.extend(new Abstract.Insertion('afterBegin'), {
  initializeRange: function() {
    this.range.selectNodeContents(this.element);
    this.range.collapse(true);
  },

  insertContent: function(fragments) {
    fragments.reverse(false).each((function(fragment) {
      this.element.insertBefore(fragment, this.element.firstChild);
    })).bind(this));
  }
});

```

```

Insertion.Bottom = Class.create();
Insertion.Bottom.prototype = Object.extend(new Abstract.Insertion('beforeEnd'), {
  initializeRange: function() {
    this.range.selectNodeContents(this.element);
    this.range.collapse(this.element);
  },

  insertContent: function(fragments) {
    fragments.each((function(fragment) {
      this.element.appendChild(fragment);
    })).bind(this));
  }
});

```

```

Insertion.After = Class.create();
Insertion.After.prototype = Object.extend(new Abstract.Insertion('afterEnd'), {
  initializeRange: function() {
    this.range.setStartAfter(this.element);
  },

  insertContent: function(fragments) {
    fragments.each((function(fragment) {
      this.element.parentNode.insertBefore(fragment,
        this.element.nextSibling);
    })).bind(this));
  }
});

```

```

});

/*-----*/

Element.ClassNames = Class.create();
Element.ClassNames.prototype = {
  initialize: function(element) {
    this.element = $(element);
  },

  _each: function(iterator) {
    this.element.className.split(/\s+/).select(function(name) {
      return name.length > 0;
    })._each(iterator);
  },

  set: function(className) {
    this.element.className = className;
  },

  add: function(classNameToAdd) {
    if (this.include(classNameToAdd)) return;
    this.set(this.toArray().concat(classNameToAdd).join(' '));
  },

  remove: function(classNameToRemove) {
    if (!this.include(classNameToRemove)) return;
    this.set(this.select(function(className) {
      return className != classNameToRemove;
    }).join(' '));
  },

  toString: function() {
    return this.toArray().join(' ');
  }
}

Object.extend(Element.ClassNames.prototype, Enumerable);
var Selector = Class.create();

```



```

var params = this.params, conditions = [], clause;

if (params.wildcard)
  conditions.push('true');
if (clause = params.id)
  conditions.push('element.id == ' + clause.inspect());
if (clause = params.tagName)
  conditions.push('element.tagName.toUpperCase() == ' + clause.inspect());
if ((clause = params.classNames).length > 0)
  for (var i = 0; i < clause.length; i++)
    conditions.push('Element.hasClassName(element, ' + clause[i].inspect() + ')');
if (clause = params.attributes) {
  clause.each(function(attribute) {
    var value = 'element.getAttribute(' + attribute.name.inspect() + ')';
    var splitValueBy = function(delimiter) {
      return value + ' && ' + value + '.split(' + delimiter.inspect() + ')';
    }

    switch (attribute.operator) {
      case '=':    conditions.push(value + ' == ' + attribute.value.inspect()); break;
      case '~=':   conditions.push(splitValueBy(' ') + '.include(' + attribute.value.inspect() +
'')); break;
      case '|=':   conditions.push(
        splitValueBy('-')      +      '.first().toUpperCase()      ==      '
        + attribute.value.toUpperCase().inspect()
      ); break;
      case '!=':   conditions.push(value + ' != ' + attribute.value.inspect()); break;
      case '':
      case undefined: conditions.push(value + ' != null'); break;
      default:     throw 'Unknown operator ' + attribute.operator + ' in selector';
    }
  });
}

return conditions.join(' && ');
},

compileMatcher: function() {
  this.match = new Function('element', 'if (!element.tagName) return false; \

```



```

        return ' + this.buildMatchExpression();
    },

    findElements: function(scope) {
        var element;

        if (element = $(this.params.id))
            if (this.match(element))
                if (!scope || Element.childOf(element, scope))
                    return [element];

        scope = (scope || document).getElementsByName(this.params.tagName || '*');

        var results = [];
        for (var i = 0; i < scope.length; i++)
            if (this.match(element = scope[i]))
                results.push(Element.extend(element));

        return results;
    },

    toString: function() {
        return this.expression;
    }
}

function $$() {
    return $A(arguments).map(function(expression) {
        return expression.strip().split(/\s+/).inject([null], function(results, expr) {
            var selector = new Selector(expr);
            return results.map(selector.findElements.bind(selector)).flatten();
        });
    }).flatten();
}

var Field = {
    clear: function() {
        for (var i = 0; i < arguments.length; i++)
            $(arguments[i]).value = "";
    },

```

```

focus: function(element) {
    $(element).focus();
},

present: function() {
    for (var i = 0; i < arguments.length; i++)
        if ($(arguments[i]).value == "") return false;
    return true;
},

select: function(element) {
    $(element).select();
},

activate: function(element) {
    element = $(element);
    element.focus();
    if (element.select)
        element.select();
}
}

/*-----*/

var Form = {
    serialize: function(form) {
        var elements = Form.getElements($(form));
        var queryComponents = new Array();

        for (var i = 0; i < elements.length; i++) {
            var queryComponent = Form.Element.serialize(elements[i]);
            if (queryComponent)
                queryComponents.push(queryComponent);
        }

        return queryComponents.join('&');
    },

```

```

getElements: function(form) {
    form = $(form);
    var elements = new Array();

    for (var tagName in Form.Element.Serializers) {
        var tagElements = form.getElementsByTagName(tagName);
        for (var j = 0; j < tagElements.length; j++)
            elements.push(tagElements[j]);
    }
    return elements;
},

```

```

getInputs: function(form, typeName, name) {
    form = $(form);
    var inputs = form.getElementsByTagName('input');

```

```

    if (!typeName && !name)
        return inputs;

```

```

    var matchingInputs = new Array();
    for (var i = 0; i < inputs.length; i++) {
        var input = inputs[i];
        if ((typeName && input.type != typeName) ||
            (name && input.name != name))
            continue;
        matchingInputs.push(input);
    }

```

```

    return matchingInputs;
},

```

```

disable: function(form) {
    var elements = Form.getElements(form);
    for (var i = 0; i < elements.length; i++) {
        var element = elements[i];
        element.blur();
        element.disabled = 'true';
    }
},

```

```

enable: function(form) {
    var elements = Form.getElements(form);
    for (var i = 0; i < elements.length; i++) {
        var element = elements[i];
        element.disabled = "";
    }
},

findFirstElement: function(form) {
    return Form.getElements(form).find(function(element) {
        return element.type != 'hidden' && !element.disabled &&
            ['input', 'select', 'textarea'].include(element.tagName.toLowerCase());
    });
},

focusFirstElement: function(form) {
    Field.activate(Form.findFirstElement(form));
},

reset: function(form) {
    $(form).reset();
}
}

Form.Element = {
    serialize: function(element) {
        element = $(element);
        var method = element.tagName.toLowerCase();
        var parameter = Form.Element.Serializers[method](element);

        if (parameter) {
            var key = encodeURIComponent(parameter[0]);
            if (key.length == 0) return;

            if (parameter[1].constructor != Array)
                parameter[1] = [parameter[1]];

            return parameter[1].map(function(value) {

```

```

        return key + '=' + encodeURIComponent(value);
    }).join('&');
}
},

getValue: function(element) {
    element = $(element);
    var method = element.tagName.toLowerCase();
    var parameter = Form.Element.Serializers[method](element);

    if (parameter)
        return parameter[1];
}
}

Form.Element.Serializers = {
    input: function(element) {
        switch (element.type.toLowerCase()) {
            case 'submit':
            case 'hidden':
            case 'password':
            case 'text':
                return Form.Element.Serializers.textarea(element);
            case 'checkbox':
            case 'radio':
                return Form.Element.Serializers.inputSelector(element);
        }
        return false;
    },

    inputSelector: function(element) {
        if (element.checked)
            return [element.name, element.value];
    },

    textarea: function(element) {
        return [element.name, element.value];
    },

```

```

select: function(element) {
  return Form.Element.Serializers[element.type == 'select-one' ?
    'selectOne' : 'selectMany'](element);
},

```

```

selectOne: function(element) {
  var value = "", opt, index = element.selectedIndex;
  if (index >= 0) {
    opt = element.options[index];
    value = opt.value || opt.text;
  }
  return [element.name, value];
},

```

```

selectMany: function(element) {
  var value = [];
  for (var i = 0; i < element.length; i++) {
    var opt = element.options[i];
    if (opt.selected)
      value.push(opt.value || opt.text);
  }
  return [element.name, value];
}
}

```

```

/*-----*/

```

```

var $F = Form.Element.getValue;

```

```

/*-----*/

```

```

Abstract.TimedObserver = function() {}
Abstract.TimedObserver.prototype = {
  initialize: function(element, frequency, callback) {
    this.frequency = frequency;
    this.element = $(element);
    this.callback = callback;

    this.lastValue = this.getValue();
  }
};

```

```

        this.registerCallback();
    },

    registerCallback: function() {
        setInterval(this.onTimerEvent.bind(this), this.frequency * 1000);
    },

    onTimerEvent: function() {
        var value = this.getValue();
        if (this.lastValue !== value) {
            this.callback(this.element, value);
            this.lastValue = value;
        }
    }
}

Form.Element.Observer = Class.create();
Form.Element.Observer.prototype = Object.extend(new Abstract.TimedObserver(), {
    getValue: function() {
        return Form.Element.getValue(this.element);
    }
});

Form.Observer = Class.create();
Form.Observer.prototype = Object.extend(new Abstract.TimedObserver(), {
    getValue: function() {
        return Form.serialize(this.element);
    }
});

/*-----*/

Abstract.EventObserver = function() {}
Abstract.EventObserver.prototype = {
    initialize: function(element, callback) {
        this.element = $(element);
        this.callback = callback;

        this.lastValue = this.getValue();

```

```

    if (this.element.tagName.toLowerCase() == 'form')
        this.registerFormCallbacks();
    else
        this.registerCallback(this.element);
},

onElementEvent: function() {
    var value = this.getValue();
    if (this.lastValue != value) {
        this.callback(this.element, value);
        this.lastValue = value;
    }
},

registerFormCallbacks: function() {
    var elements = Form.getElements(this.element);
    for (var i = 0; i < elements.length; i++)
        this.registerCallback(elements[i]);
},

registerCallback: function(element) {
    if (element.type) {
        switch (element.type.toLowerCase()) {
            case 'checkbox':
            case 'radio':
                Event.observe(element, 'click', this.onElementEvent.bind(this));
                break;
            case 'password':
            case 'text':
            case 'textarea':
            case 'select-one':
            case 'select-multiple':
                Event.observe(element, 'change', this.onElementEvent.bind(this));
                break;
        }
    }
}
}

```



```

Form.Element.EventObserver = Class.create();
Form.Element.EventObserver.prototype = Object.extend(new Abstract.EventObserver(), {
  getValue: function() {
    return Form.Element.getValue(this.element);
  }
});

```

```

Form.EventObserver = Class.create();
Form.EventObserver.prototype = Object.extend(new Abstract.EventObserver(), {
  getValue: function() {
    return Form.serialize(this.element);
  }
});
if (!window.Event) {
  var Event = new Object();
}

```

```

Object.extend(Event, {
  KEY_BACKSPACE: 8,
  KEY_TAB:      9,
  KEY_RETURN:   13,
  KEY_ESC:      27,
  KEY_LEFT:     37,
  KEY_UP:       38,
  KEY_RIGHT:    39,
  KEY_DOWN:     40,
  KEY_DELETE:   46,

  element: function(event) {
    return event.target || event.srcElement;
  },

  isLeftClick: function(event) {
    return (((event.which) && (event.which == 1)) ||
      ((event.button) && (event.button == 1)));
  },

  pointerX: function(event) {
    return event.pageX || (event.clientX +

```

```

    (document.documentElement.scrollLeft || document.body.scrollLeft));
},

pointerY: function(event) {
    return event.pageY || (event.clientY +
        (document.documentElement.scrollTop || document.body.scrollTop));
},

stop: function(event) {
    if (event.preventDefault) {
        event.preventDefault();
        event.stopPropagation();
    } else {
        event.returnValue = false;
        event.cancelBubble = true;
    }
},

// find the first node with the given tagName, starting from the
// node the event was triggered on; traverses the DOM upwards
findElement: function(event, tagName) {
    var element = Event.element(event);
    while (element.parentNode && (!element.tagName ||
        (element.tagName.toUpperCase() != tagName.toUpperCase())))
        element = element.parentNode;
    return element;
},

observers: false,

_observeAndCache: function(element, name, observer, useCapture) {
    if (!this.observers) this.observers = [];
    if (element.addEventListener) {
        this.observers.push([element, name, observer, useCapture]);
        element.addEventListener(name, observer, useCapture);
    } else if (element.attachEvent) {
        this.observers.push([element, name, observer, useCapture]);
        element.attachEvent('on' + name, observer);
    }
}

```

```

},

unloadCache: function() {
  if (!Event.observers) return;
  for (var i = 0; i < Event.observers.length; i++) {
    Event.stopObserving.apply(this, Event.observers[i]);
    Event.observers[i][0] = null;
  }
  Event.observers = false;
},

observe: function(element, name, observer, useCapture) {
  var element = $(element);
  useCapture = useCapture || false;

  if (name == 'keypress' &&
      (navigator.appVersion.match(/Konqueror|Safari|KHTML/)
       || element.attachEvent))
    name = 'keydown';

  this._observeAndCache(element, name, observer, useCapture);
},

stopObserving: function(element, name, observer, useCapture) {
  var element = $(element);
  useCapture = useCapture || false;

  if (name == 'keypress' &&
      (navigator.appVersion.match(/Konqueror|Safari|KHTML/)
       || element.detachEvent))
    name = 'keydown';

  if (element.removeEventListener) {
    element.removeEventListener(name, observer, useCapture);
  } else if (element.detachEvent) {
    element.detachEvent('on' + name, observer);
  }
}
});

```

```

/* prevent memory leaks in IE */
if (navigator.appVersion.match(/bMSIE\b/))
    Event.observe(window, 'unload', Event.unloadCache, false);
var Position = {
    // set to true if needed, warning: firefox performance problems
    // NOT needed for page scrolling, only if draggable contained in
    // scrollable elements
    includeScrollOffsets: false,

    // must be called before calling withinIncludingScrolloffset, every time the
    // page is scrolled
    prepare: function() {
        this.deltaX = window.pageXOffset
            || document.documentElement.scrollLeft
            || document.body.scrollLeft
            || 0;
        this.deltaY = window.pageYOffset
            || document.documentElement.scrollTop
            || document.body.scrollTop
            || 0;
    },

    realOffset: function(element) {
        var valueT = 0, valueL = 0;
        do {
            valueT += element.scrollTop || 0;
            valueL += element.scrollLeft || 0;
            element = element.parentNode;
        } while (element);
        return [valueL, valueT];
    },

    cumulativeOffset: function(element) {
        var valueT = 0, valueL = 0;
        do {
            valueT += element.offsetTop || 0;
            valueL += element.offsetLeft || 0;
            element = element.offsetParent;
        } while (element);
    }
};

```

```

    } while (element);
    return [valueL, valueT];
},

```

```

positionedOffset: function(element) {
    var valueT = 0, valueL = 0;
    do {
        valueT += element.offsetTop || 0;
        valueL += element.offsetLeft || 0;
        element = element.offsetParent;
        if (element) {
            p = Element.getStyle(element, 'position');
            if (p == 'relative' || p == 'absolute') break;
        }
    } while (element);
    return [valueL, valueT];
},

```

```

offsetParent: function(element) {
    if (element.offsetParent) return element.offsetParent;
    if (element == document.body) return element;

    while ((element = element.parentNode) && element != document.body)
        if (Element.getStyle(element, 'position') != 'static')
            return element;

    return document.body;
},

```

```

// caches x/y coordinate pair to use with overlap
within: function(element, x, y) {
    if (this.includeScrollOffsets)
        return this.withinIncludingScrolloffsets(element, x, y);
    this.xcomp = x;
    this.ycomp = y;
    this.offset = this.cumulativeOffset(element);

    return (y >= this.offset[1] &&
        y < this.offset[1] + element.offsetHeight &&

```

```

        x >= this.offset[0] &&
        x < this.offset[0] + element.offsetWidth);
    },

    withinIncludingScrolloffsets: function(element, x, y) {
        var offsetcache = this.realOffset(element);

        this.xcomp = x + offsetcache[0] - this.deltaX;
        this.ycomp = y + offsetcache[1] - this.deltaY;
        this.offset = this.cumulativeOffset(element);

        return (this.ycomp >= this.offset[1] &&
            this.ycomp < this.offset[1] + element.offsetHeight &&
            this.xcomp >= this.offset[0] &&
            this.xcomp < this.offset[0] + element.offsetWidth);
    },

    // within must be called directly before
    overlap: function(mode, element) {
        if (!mode) return 0;
        if (mode == 'vertical')
            return ((this.offset[1] + element.offsetHeight) - this.ycomp) /
                element.offsetHeight;
        if (mode == 'horizontal')
            return ((this.offset[0] + element.offsetWidth) - this.xcomp) /
                element.offsetWidth;
    },

    clone: function(source, target) {
        source = $(source);
        target = $(target);
        target.style.position = 'absolute';
        var offsets = this.cumulativeOffset(source);
        target.style.top = offsets[1] + 'px';
        target.style.left = offsets[0] + 'px';
        target.style.width = source.offsetWidth + 'px';
        target.style.height = source.offsetHeight + 'px';
    },

```

```

page: function(forElement) {
    var valueT = 0, valueL = 0;

    var element = forElement;
    do {
        valueT += element.offsetTop || 0;
        valueL += element.offsetLeft || 0;

        // Safari fix
        if (element.offsetParent==document.body)
            if (Element.getStyle(element,'position')=='absolute') break;

    } while (element = element.offsetParent);

    element = forElement;
    do {
        valueT -= element.scrollTop || 0;
        valueL -= element.scrollLeft || 0;
    } while (element = element.parentNode);

    return [valueL, valueT];
},

clone: function(source, target) {
    var options = Object.extend({
        setLeft: true,
        setTop: true,
        setWidth: true,
        setHeight: true,
        offsetTop: 0,
        offsetLeft: 0
    }, arguments[2] || {})

    // find page position of source
    source = $(source);
    var p = Position.page(source);

    // find coordinate system to use
    target = $(target);

```

```

var delta = [0, 0];
var parent = null;
// delta [0,0] will do fine with position: fixed elements,
// position:absolute needs offsetParent deltas
if (Element.getStyle(target,'position') == 'absolute') {
    parent = Position.offsetParent(target);
    delta = Position.page(parent);
}

// correct by body offsets (fixes Safari)
if (parent == document.body) {
    delta[0] -= document.body.offsetLeft;
    delta[1] -= document.body.offsetTop;
}

// set position
if(options.setLeft) target.style.left = (p[0] - delta[0] + options.offsetLeft) + 'px';
if(options.setTop) target.style.top = (p[1] - delta[1] + options.offsetTop) + 'px';
if(options.setWidth) target.style.width = source.offsetWidth + 'px';
if(options.setHeight) target.style.height = source.offsetHeight + 'px';
},

absolutize: function(element) {
    element = $(element);
    if (element.style.position == 'absolute') return;
    Position.prepare();

    var offsets = Position.positionedOffset(element);
    var top = offsets[1];
    var left = offsets[0];
    var width = element.clientWidth;
    var height = element.clientHeight;

    element._originalLeft = left - parseFloat(element.style.left || 0);
    element._originalTop = top - parseFloat(element.style.top || 0);
    element._originalWidth = element.style.width;
    element._originalHeight = element.style.height;

    element.style.position = 'absolute';

```



```

    element.style.top    = top + 'px';
    element.style.left   = left + 'px';
    element.style.width  = width + 'px';
    element.style.height = height + 'px';
  },

  relativize: function(element) {
    element = $(element);
    if (element.style.position == 'relative') return;
    Position.prepare();

    element.style.position = 'relative';
    var top  = parseFloat(element.style.top  || 0) - (element._originalTop || 0);
    var left = parseFloat(element.style.left || 0) - (element._originalLeft || 0);

    element.style.top    = top + 'px';
    element.style.left   = left + 'px';
    element.style.height = element._originalHeight;
    element.style.width  = element._originalWidth;
  }
}

// Safari returns margins on body which is incorrect if the child is absolutely
// positioned.  For performance reasons, redefine Position.cumulativeOffset for
// KHTML/WebKit only.
if (/Konqueror|Safari|KHTML/.test(navigator.userAgent)) {
  Position.cumulativeOffset = function(element) {
    var valueT = 0, valueL = 0;
    do {
      valueT += element.offsetTop || 0;
      valueL += element.offsetLeft || 0;
      if (element.offsetParent == document.body)
        if (Element.getStyle(element, 'position') == 'absolute') break;

      element = element.offsetParent;
    } while (element);

    return [valueL, valueT];
  }
}

```

}

## APÊNDICE II

```
CREATE TABLE Garcom (  
  Gar_ID INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Gar_Nome VARCHAR(40) NULL,  
  PRIMARY KEY(Gar_ID)  
)  
TYPE=InnoDB;
```

```
CREATE TABLE Produto (  
  Prod_ID INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Prod_Nome VARCHAR(40) NULL,  
  Prod_Precos DOUBLE(3,2) NOT NULL,  
  Prod_Tempo_Preparo INT NULL,  
  Prod_Estoque INTEGER UNSIGNED NULL,  
  Prod_Tipo CHAR NULL,  
  PRIMARY KEY(Prod_ID)  
)  
TYPE=InnoDB;
```

```
CREATE TABLE Cliente (  
  Cli_ID INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Cli_Telefone INTEGER(10) UNSIGNED NOT NULL,  
  Cli_Nome VARCHAR(40) NULL,  
  Cli_Dt_Nascimento DATE NULL,  
  PRIMARY KEY(Cli_ID),  
  UNIQUE INDEX Cliente_tel_index(Cli_Telefone)  
)  
TYPE=InnoDB;
```

```
CREATE TABLE Usuario (  
  Usr_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Usr_login VARCHAR(10) NULL,  
  Usr_senha VARCHAR(120) NULL,  
  Usr_perfil INTEGER UNSIGNED NULL,  
  PRIMARY KEY(Usr_id)  
)  
TYPE=InnoDB;
```

```

CREATE TABLE Mesa (
Mesa_ID INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
Garcom_Gar_ID INTEGER UNSIGNED NULL,
Mesa_Nome VARCHAR(15) NULL,
Mesa_Dt_escolhida DATE NULL,
PRIMARY KEY(Mesa_ID),
INDEX Mesa_Garcom_FK(Garcom_Gar_ID),
FOREIGN KEY(Garcom_Gar_ID)
REFERENCES Garcom(Gar_ID)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
TYPE=InnoDB;

```

```

CREATE TABLE Conta (
Cta_ID INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
Cli_ID INTEGER UNSIGNED NOT NULL,
Mesa_ID INTEGER UNSIGNED NOT NULL,
Cta_Data_Ini TIMESTAMP NULL,
Cta_Data_Final TIMESTAMP NULL,
Cta_Status CHAR BINARY NULL,
PRIMARY KEY(Cta_ID),
INDEX Conta_Cliente_FK(Cli_ID),
INDEX Conta_Mesa_FK(Mesa_ID),
FOREIGN KEY(Cli_ID)
REFERENCES Cliente(Cli_ID)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
FOREIGN KEY(Mesa_ID)
REFERENCES Mesa(Mesa_ID)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
TYPE=InnoDB;

```

```

CREATE TABLE Pedido (
Ped_ID INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
Cta_ID INTEGER UNSIGNED NOT NULL,

```

```

Ped_Estado CHAR BINARY NULL,
Ped_Data_Reg TIMESTAMP NULL,
Ped_Data_Fim TIMESTAMP NULL,
PRIMARY KEY(Ped_ID),
INDEX Pedido_Conta_FK(Cta_ID),
FOREIGN KEY(Cta_ID)
REFERENCES Conta(Cta_ID)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
TYPE=InnoDB;

```

```

CREATE TABLE Pedido_has_Produto (
Ped_ID INTEGER UNSIGNED NOT NULL,
Prod_ID INTEGER UNSIGNED NOT NULL,
Has_Qtde INTEGER UNSIGNED NULL,
Has_Andamento CHAR BINARY NULL,
PRIMARY KEY(Ped_ID, Prod_ID),
INDEX Pedido_Produto_FK(Ped_ID),
INDEX Produto_Pedido_FK(Prod_ID),
FOREIGN KEY(Ped_ID)
REFERENCES Pedido(Ped_ID)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
FOREIGN KEY(Prod_ID)
REFERENCES Produto(Prod_ID)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
TYPE=InnoDB;

```

## APÊNDICE III

```

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class VerificarPedido{
    /**
     * Este método irá realizar a gravação dos dados no arquivo texto
     * @throws IOException
     */
    public void executarVerificacao() throws IOException {
        /**
         * objeto do tipo FileOutputStream encarregado por abrir/escrever/fechar o arquivo texto
         */
        FileOutputStream fous = null;
        /**
         * objeto do tipo PrintStream encarregado por escrever no arquivo texto
         */
        PrintStream ps = null;
        try {
            /**
             * Cria um novo StringBuffer de retorno
             */
            StringBuffer retorno = new StringBuffer();
            /**
             * Cria uma lista com os pedidos ativos. Ver método <code>retornaPedidoAtivos</code>
             */

            List listaPedidos = retornaPedidoAtivos();
            if (listaPedidos != null && listaPedidos.size() > 0) {

```

```

        for (int i = 0; i < listaPedidos.size(); i++) {
            /**
             * Inclui um item da lista no StringBuffer
             */
            retorno.append(listaPedidos.get(i) + "\n");
        }
    } else {
        retorno.append("NAO EXISTEM PEDIDOS");
    }
    /**
     * cria o arquivo no diretório c:/
     */
    fous = new FileOutputStream("c://arquivo.txt");
    /**
     * realiza a criação do stream com o arquivo criado
     */
    ps = new PrintStream(fous);
    /**
     * imprime no arquivo a string(StringBuffer) de retorno
     */
    ps.print(retorno);
} catch (FileNotFoundException e) {
    e.printStackTrace();
} finally {
    ps.close();
    fous.close();
}
}
/**
 * Este método será responsável por buscar no banco de dados o primeiro
 * pedido ativo e retornar um array de String com todos os produtos e suas
 * quantidades. Ex: Filé(1)
 *
 * @return List
 */
public List retornaPedidoAtivos() {
    List listaPedidos = null;
    Connection con = null;
    Statement stmt = null;

```

```

ResultSet rs = null;
try {
    /**
     * Registra no DriverManager o Driver do Mysql
     */
    Class.forName("com.mysql.jdbc.Driver");
    /**
     * cria a conexão através do DriverManager passando servidor, login
     * e senha
     */
    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mannu", "root", "");
    /**
     * cria o statement que executará a consulta
     */
    stmt = con.createStatement();
    /**
     * cria um ArrayList para incluir os pedidos
     */
    listaPedidos = new ArrayList();
    /**
     *
     * Esta segunda consulta foi criada por causa da explicação
     * acima.
     *
     */
    String sqlSelecionaProdutosPorPedido = "SELECT prod.Prod_nome, php.Has_Qtde as QTDE "+
        "FROM Conta cta, Pedido ped, Pedido_has_Produto php, Produto prod "+
        "WHERE cta.Cta_Id = ped.Cta_Id "+
        "AND ped.Ped_ID = php.Ped_ID "+
        "AND php.Prod_ID = prod.Prod_ID "+
        "AND prod.Prod_Tipo = 0 "+
        "AND cta.Cta_Status = 'A' "+
        "AND ped.Ped_Estado = 'A' "+
        "order by ped.Ped_Data_Reg asc, prod.Prod_nome asc";
    /**
     * executa uma nova consulta e retorna um ResultSet para
     * manipular um dado
     */
    rs = stmt.executeQuery(sqlSelecionaProdutosPorPedido);

```



```

    /**
     * Enquanto existirem registro, inclui no ArrayList os produtos
     * e as suas quantidades
     */
    int ordem = 1;
    while (rs.next()) {
        String nomeProduto = rs.getString("Prod_nome");
        int qtde = rs.getInt("QTDE");
        listaPedidos.add(ordem + " - " + nomeProduto + "(" + qtde + ")");
        ordem++;
    }
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    try {
        if (rs != null)
            rs.close();
        if (stmt != null)
            stmt.close();
        if (con != null)
            con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
return listaPedidos;
}

```

```

public static void main(String args[]) {
    VerificarPedido vp = new VerificarPedido();
    /**
     * Cria uma nova Thread passando a classe atual que implementa Runnable
     */
    while (true) {
        try {

```

```
        vp.executarVerificacao();  
    /**  
    * realiza uma pausa de 5 segundos para que possa ser re-executado este método  
    */  
        Thread.sleep(5000);  
    }  
    /**  
    * se lançar excessão não faça nada e tente novamente  
    */  
    catch (Exception e) {}  
    }  
}
```